

Chapter 7

Contextual Recommender Systems in Business from Models to Experiments



Khedija Arour and Rim Dridi

7.1 Introduction

Recommender Systems (RS) are information filtering systems that cope with the information overload problem by filtering vital information fragment out of large amount of dynamically generated information according to users preferences about items. These systems try to give solutions to resolve this problem by searching through large volume of existing information to provide users with adapted content and services [43]. Instead of exploring an important number of items until finding the most adequate, RS became a promising area of research, thanks to their help for users to suggest the items they might prefer. However, there are usually various factors that may impact users' preferences. Therefore, research in recommender systems is starting to recognize the importance of items and the role of user's context in enhancing the recommendation output. In this respect, traditional recommender systems are extended to offer novel lines of research areas such as Context-Aware Recommender Systems (CARS). This chapter provides a survey on CARS systems and presents a standard evaluation process that can be adopted by researches on this field. We review a range of evaluation metrics and measures as well as some approaches used for evaluating recommender systems.

The layout of this chapter is structured as follows. In Sect. 7.2, the different notions and concepts are presented. Section 7.3 investigates context awareness recommender systems in detail. Evaluation process for CARS systems is given in

K. Arour (✉)
College of Business, University of Jeddah, Jeddah, Saudi Arabia
e-mail: kbarour@uj.edu.sa

R. Dridi
LISI Laboratory, INSAT, University of Carthage, Tunis, Tunisia
e-mail: rim.dridi@ensi-uma.tn

Sect. 7.4. Experimental results are detailed in Sect. 7.5. In Sect. 7.6, some CARS systems applied for specific business are presented. We conclude and give some future directions in Sect. 7.7.

7.2 Recommender Systems: Notions and Concepts

Some notions and concepts related to recommender systems must be presented. For example, the user, the *Item* that should be recommended, and the *Rating* that represent how much a user likes an item. So the triplet (User, Item, Rating) is the core of any recommender systems.

- **User:** this term depicts the set of entities to which recommendations will be given, regardless of whether they describe a person, a group of people, or other entities of interest. Generally, users designate the persons to whom items are suggested, generally presented using attributes such as the id, name, gender, age, etc. These information are modelled as “user profile” aiming to identify the user’s needs for providing custom recommendations which could be suitable for the user. Ordinary users having a sufficient number of ratings have been distinguished from particular ones who require a special reasoning to satisfy all users’ needs. In this regard, three types of particular users are identified [22]: (i) “cold start users” are the new users recently entered the system with very limited information (insufficient ratings); (ii) “grey sheep users” are the users with unusual tastes resulting low correlations with other users; and (iii) users who do not have any behavior in the current context.
- **Item:** the items are objects to be recommended to users, regardless of their actual representation. Generally, typical recommended items are documents, music, movies, etc. An item can be characterized by its features or descriptions and utility (positive if it is beneficial for the user and negative if not) [43]. In particular, the paper in [43] describes an item in a movie recommender system through the following attributes: title, length, genre, director, and release year.
- **Rating:** we denote the preference of a user toward an item as a rating. In our study of recommendation systems, we take user’s rating to be the quintessential piece of information utilized to indicate a user’s interest in an item. From this point of view, the rating presents the interaction between a user and the recommender system aiming to infer the user’s opinions. We equate higher ratings with a greater preference (i.e., users would like better an item rated 5 rather than an item rated 2). A rating can be viewed in different forms: (i) *binary rating* that shows whether a given item is good for a user or not. As an exemplification, in YouTube¹ “like” and “follow” could be considered as a binary rating. While, binary rating is easy for the user to deal with and less ambiguous, it cannot be sufficient for

¹ <https://www.youtube.com>.

items comparison; (ii) *numerical rating* uses a numerical scale rating aiming to provide detailed feedback. Take Netflix as an example, it uses standard five-star rating scale to power its review system and recommendations. There are also variations like using a ten-star scale; and (iii) *ordinal rating* are basically used to clarify the meaning of each rating level with words such as 1/5 stars means “I do not like very much” and 5/5 stars means “I really like” [18].

7.2.1 Foundations of Recommender Systems

Let us first look at comprehending what a recommender system is and what types of functionalities do recommender systems have.

The concept of recommender systems was first introduced by Resnick and Varian in 1997 [41]. Indeed, the developers of the first recommender named Tapestry [20] have considered their system as a collaborative filtering system. Yet, the authors in [41] have chosen the term "recommender system" for these reasons: (i) recommenders and recipients may be unknown to each other, then may not explicitly collaborate; and (ii) recommendations may propose some particularly pertinent items, as well indicating those that should be filtered. Hence, they have considered recommender systems as an independent research area in the mid-1990s issued from different other areas such as information retrieval, approximation theory, management sciences, and also cognitive science.

Firstly, we start with a simple definition of a recommender system: A recommender system is a system able to suggest items to users [41]. More abstractly, a recommender system is a system that suggests content a user is interested in out of an enormous set of choices [41] and hence, is a system to overcome the information overload problem. For this task, a recommender system aims at predicting the most relevant items to a user and states a short list of recommendations. According to this definition, we derive two main tasks: (i) the rating prediction task and the (ii) top-n recommendations task. The latter task is based on the first one, as a recommender system orders the list of recommended items by the predicted rating representing the perceived usefulness of a user towards an item.

In another point of view, authors in [11] have differentiated between recommender systems and information retrieval systems by the power of recommenders to be personalized in addition to their ability to suggest relevant recommendations. Therefore, they propose the following definition: “A recommender system is any system that produces individualized recommendations as output or has the effect of guiding the user, in a personalized way, to interesting or useful objects in a large space of possible options” [11]. Authors assumed in their article [34] that recommendation is related to four main features. These features are important because they cover the necessary needs of users facing many set of items: Decide, Compare, Explore, and Discover. As recommender systems has been developed in different industrial domains, like: e-commerce, health-care, entertainment, etc. Many works have proposed their definition according to the application field

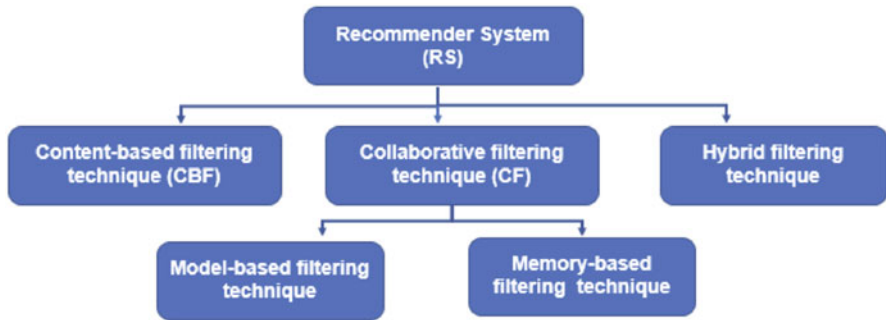


Fig. 7.1 RS classification

particularities. For instance, in e-commerce discipline, researches in [40] have considered recommender systems as computer algorithms used widely in e-commerce to propose items to a user, like what items to buy, news to read, or movies to rent.

7.2.2 Classification of Recommender Systems

Recommendation approaches are expected to predict the utilities of items for target users and offer accurate recommendations. It is possible to classify RS approaches by various ways in accordance with different criteria including the type of feedback they use (explicit or implicit feedback), the recommendation task they address (rating prediction or top-N recommendation), etc. The most common classification used in the literature is based on the type of data exploited for recommendation and establishes the following three categories (see Fig. 7.1):

- *Content-Based Filtering (CBF) approaches.* These approaches make use of knowledge related to users or items to provide recommendation.
- *Collaborative Filtering (CF) approaches.* These approaches recommend items relying on similar users and their ratings.
- *Hybrid approaches.* These approaches combine the two above-mentioned filtering approaches.

7.2.2.1 Content-Based Filtering

Content-based recommender systems are based on content information about users or items to provide recommendations. This information can take different forms like features, textual descriptions, and tags. In other words, users receive items suggestions that are similar to those they positively evaluated in the past. Particularly, recommendations are made through matching the user profile features describing the

user's preferences with the items features. In content-based recommender systems, the item can be represented by a weighted terms vector extracted from its content. To define the user profile, CBF mostly concentrate on the model of the user's preference or the history of the user's interaction with the recommender.

Pandora Music Genome Project² is an example of a content-based approach that uses the characteristics of a song or a singer (subset of attributes describing songs) to capture the essence of music with similar characteristics and to organize them. Users' feedbacks (likes or dislikes) are adopted to filter the music station's results. Basically, a content-based recommender system comprises the following steps [32]:

1. Preprocessing of items content (e.g., Web pages, documents, product descriptions, etc.) to extract structured pertinent information (e.g., Web pages represented as keyword vectors).
2. Starting from items liked or disliked in the past, the profile of a target user is learned through machine learning techniques.
3. Matching the profile representation of the target user and that of items to be recommended computed using similarity metrics.
4. Recommending a ranked list of potentially pertinent items.

This technique presents advantages such as user independence, since CBF systems only use ratings of the active user to build the recommendation model. Additionally, when a new item appears and has not yet been rated, CBF systems are able to recommend it. However, CBF suffer from several issues such as the over-specialization, as they are not capable of finding unexpected items: the user will receive recommendations of items similar to the ones rated before.

7.2.2.2 Collaborative Filtering

To date, collaborative filtering (CF) is the most popular algorithm used to design various applications and sites for recommender systems such as Facebook,³ Twitter,⁴ Google,⁵ LinkedIn,⁶ and Netflix. The underlying idea behind CF is that users with common interests in the past are more likely to keep exhibiting similar interests in the future. The principal property to work with collaborative filtering are the ratings given by users for items. Therefore, the typical input of collaborative recommender systems is represented by a matrix of ratings representing users by rows and items by columns. More precisely, the user-item matrix defining users' preferences for items is used to find like-minded users by computing similarities between their profiles

² <https://www.pandora.com>.

³ <https://facebook.com/>.

⁴ <https://twitter.com>.

⁵ <https://www.google.com>.

⁶ <https://fr.linkedin.com>.

defining a “neighborhood” to provide recommendations. In general, a collaborative filtering system requires the following steps to generate recommendations:

1. Identification of the subject of the recommendation (ratings of the target user).
2. Identification of the most similar users to the target one using a similarity function (cosine similarity, Pearson’s correlation, etc.).
3. Identification of the rated items by the similar users and not rated by the target one.
4. Prediction of the rating of each selected item based on users’ similarity.
5. Recommendation of items according to the predicted ratings.

There are two main recommendation techniques in collaborative filtering: memory-based and model-based algorithms.

- **Memory-based algorithms:**
The memory-based approach uses the entire user-item matrix to find similarities between users for estimating rating predictions. It is commonly referred to as neighborhood-based or heuristic-based approach. This approach uses previous users ratings for predicting ratings for new items using one of these two ways: user-based CF recommendation or item-based CF recommendation.
- **Model-based algorithms:**
The model-based approach uses a collection of ratings in a learning phase, in which a model of user preferences is built to make intelligent rating predictions based on the observed data. Model-based CF algorithms are developed using data mining techniques and machine learning algorithms such as Bayesian networks, clustering, neural networks, linear regression and latent factor models. These latter models are known as prevalent since they use latent variables in order to explain user preferences and perform a dimensionality reduction of the rating matrix for recommendation purposes.

7.2.2.3 Hybrid Approaches

The hybrid filtering recommendation system is a system that associates two or more recommendation techniques for better recommendation performance. As stated by Burke [11], a hybrid recommender system combines multiple techniques together to obtain some synergy between them. Hybrid recommender systems have been proposed to overcome the weaknesses of collaborative filtering and content-based algorithms by combining them together instead of using them separately. This trend had also been affected in competitions such as the Netflix Prize,⁷ where the winning candidate highlighted the fact that better results are often obtained when different recommendation algorithms are associated in a single model [7].

⁷ <http://www.netflixprize.com>.

7.3 Context Awareness Recommender Systems

The use of contextual information is considered as a key component to boost the performance of systems that fall within numerous research disciplines, like mobile computing, information retrieval and recommender systems [14, 48]. In fact, the contextual information illustrated through different factors makes it possible to afford the most relevant information to the user when it is most needed. In what follows, we define the basic concepts of context and the notions that it entails.

7.3.1 Definitions

Due to the complexity and the wideness of the context concept, it has no a single definition. Indeed, context is a multifaceted concept that has been studied in various research fields and many gave multiple definitions, often different from the others and more specified than the general dictionary definition which describe context as: “conditions or circumstances that have an effect on something”. Given the growing importance of context, an entire conference, CONTEXT,⁸ is devoted for presenting and discussing this topic in wide range of various disciplines including artificial intelligence, cognitive science, linguistics, philosophy, and psychology. Based on a general point of view, the majority of renowned dictionaries have defined the context by almost similar definitions.

According to Oxford Advanced Learner’s Dictionary,⁹ “a context is the situation in which something happens and that helps you to understand it”. WordNet Search 3.1¹⁰ considers a context as “the set of facts or circumstances that surround a situation or event”. For Cambridge dictionary,¹¹ the context is viewed as “the situation within which something exists or happens, and that can help explain it”. Moreover, In Webster’s dictionary¹² “a context is defined as the interrelated conditions in which something exists or occurs like environment and setting”.

More specifically than the dictionaries definitions, many researchers presented and discussed several context definitions from different fields. The idea of including context in computer sciences was introduced in 1994 by Schilit [48], which defined the context as: “*location and the identity of nearby people and objects*”. In accordance with Schilit, “context encompasses more than just user’s location, because other things of interest are also mobile and changing”. Context could also include lighting, noise level, communication bandwidth, network connectivity and

⁸ <http://context-07.ruc.dk>.

⁹ <http://www.oxfordlearnersdictionaries.com/>.

¹⁰ <http://wordnetweb.princeton.edu/perl/webwn>.

¹¹ <http://dictionary.cambridge.org/dictionary/>.

¹² <https://www.merriam-webster.com/dictionary/>.

even the social situation (e.g., whether you are with your manager or with a co-worker). Later, a more abstract definition [16] presented by Dey and Abowd in 1999 states that: *context is defined as any information that can be used to characterize the situation of entities (place, people, and things), including the user and application and the interaction between them.* This is probably the most commonly and widely used definition for context in the computational sciences.

7.3.2 Context-Aware Recommender Systems Approaches

The recommendation field is one branch that adopted contextual information allowing recommender systems to be mightily contextualized to enhance the way in which these systems work.

With the goal of understanding the state of the art of this field, we provide a thorough literature review which analyses relevant Context-Aware Recommender Systems (CARS) approaches along several application domains, context types, recommendation techniques and paradigm for incorporating context.

In our discussion, we will use the term *contextual dimension* referring to a contextual factor (e.g., weather, time, etc.). The term *contextual condition* refers to a specific value in a contextual dimension (e.g., rainy, morning).

Among the earliest works on context-aware recommendation, the one proposed by Adomavicius et al. [1], who built a multidimensional recommendation model by integrating additional contextual dimensions besides the typical information on users and items. For rating prediction, this approach applied the collaborative filtering technique.

Since the early works on context-aware recommender systems, there have been many efforts made in this field where researchers have often tried to make use of contextual information to enhance standard recommendation algorithms. These recommendation approaches can generally be sub-divided by the formation of the utility function into memory-based and model-based approaches.

In the literature, many attempts have been made in order to build context-aware recommendation systems by applying memory-based methods. Two primary types of memory based have been introduced: the user-based, which builds neighbors according to users similarity; and the item-based, which constructs neighbors depending on items similarity. Typical examples of these approaches are the neighborhood-based collaborative filtering. In this respect, Lamche and co-workers [28], proposed and evaluated a context-aware recommender system in a mobile shopping scenario. It employed the nearest neighbor algorithm to recommend pertinent items according to the relevant selected contextual dimensions. For the task of Point-of-Interest (POI) recommendation, authors in [52] integrated the spatial, temporal, and the social context in their recommendation model. They exploited various contextual dimensions in a collaborative filtering algorithm by varying their weights to investigate the effect of including each dimension on recommendation accuracy. Otebolaku et al. [39] proposed an approach that

emphasizes the importance of similarity between contextual dimensions. To predict user preferences, K-nearest neighbors (KNN) algorithm was adopted based on the similarity between user contexts and those of other users.

It is believed that there is still a space to enhance memory-based approaches, in order to compete with the model-based approaches. In particular, several efforts followed the evolution of model-based approaches to adapt them for context-aware recommendation. Therefore, many extended models of Matrix Factorization (MF) technique were proposed in the literature, like the contextual matrix factorization, also known as Context-Aware Matrix Factorization (CAMF). It was initially introduced in [4] to model the relatedness between the contexts and item ratings providing additional model parameters. Along with standard CAMF recommender systems, we investigate more recent CAMF researches. In [23] authors proposed a context-aware latent factor model realized using matrix factorization. This study integrated contextual information of both user and item in the absence of the historical user or item data to perform event recommendations.

However, the majority of the surveyed CAMF recommendation methods cannot fully capture the impact of the relevant contextual dimensions as well as their associations on the predicted rating. To tackle this shortcoming, an improved CAMF recommendation model on the basis of the fuzzy measures of contextual dimensions [17] was proposed. It consists of two strategies extended from the correlation based CAMF-MCS model suggested in [56]. Both of the two strategies apply a common rating prediction formula given by Zheng [56], highlighting the notion of “contextual correlation”.

Besides matrix factorization-based latent factor models, others model-based algorithms have been receiving attention counting on multidisciplinary techniques such as machine learning and deep learning. These techniques have revolutionized the data mining and information retrieval techniques offering an effective impact on context-aware recommendation. For example, in [2], authors built context-aware local recommendation models where users were clustered, regarding visited destinations each period of the year. Here, the k-means clustering technique is applied to generate k clusters of countries where residents have similar behaviors according to their country of residence and to the visited destinations in different periods of visits. In reference [47], a context-aware smartphone application was developed based on artificial intelligence mechanisms to reduce the large dimensionality of context data. The principal component analysis was considered for dimensionality reduction and decision tree for building the prediction model.

7.3.3 Context-Aware Recommender Systems: Synthesis

The majority of the existing CARS follow the common classification that exists for the traditional RS: collaborative filtering, content-based filtering and hybrid recommendation approaches. That means that these works did not invent a new specific

classification for CARS. In these approaches, the context is often integrated directly into the recommendation model when it is used for producing recommendations.

Another important aspect of the literature is the widespread interest in using collaborative filtering approaches, which play a principal role in the success of several CARS [29]. These recommendation systems only depend on the user past behavior. Contrary to content-based approaches which require additional information about items. In CF approaches, the most widely used algorithms are the model-based considering users ratings to build a learning model.

The matrix factorization methods are the most employed in the model-based approaches. In the presented approaches, several variations and extensions of MF methods have been used. Model-based algorithms were developed using different machine learning techniques where a recommendation approach can be viewed as a classification problem to identify what might interest the user and what might not. Various algorithms are used for this task, such as decision trees [46, 47] and clustering [24, 58].

Despite the popularity of the research around CARS, some of the existing studies still mainly rely on incomplete assumptions about how to work with contextual information. Many CARS [24, 33] assumed that all existing contextual dimensions have equal effects and should contribute to make recommendations. Some studies [31, 46] mainly focused on the approach's research area and assumed that common contextual dimensions could be selected as relevant in compliance with their application domain. Although plenty of solutions have been proposed for the problems in the area of context-aware recommendation, the majority of them represents distinct methods for discovering relevant or correlated contextual dimensions. The lack of methods that deal with both contextual information relevancy and correlation is a quite challenging process. We believe that it is essential to combine these two topics to be handled by one method for mitigating the computation complexity and the dimensionality of context representation.

7.4 Experimental Evaluation Process for CARS Systems

Evaluation is a systematic determination of merit, worth, and significance [35]. We can measure some aspects like how accurate is a recommendation?, how many users are satisfied with the system?, does the system have an impact on user actions/reactions? does the system have an impact on business value?,...

Hence, we need to identify the role of the recommender system in the business to maximize the system utility like the time on site, the profit, etc. Also, to be able to predict the rating that a user will assign an item then to predict the best recommendation.

Evaluation process plays an important role in the context of comparative evaluation of any RS or CARS systems. It is clear that the performance of any recommender system is based heavily on data. They make reliable recommendations based on the facts that they have. Also, it is important to define appropriate

evaluation methodologies and metrics to measure the weakness and strength of the compared approaches.

Any RS or CARS paper claims that System X is better than System Y in terms of an effectiveness or efficiency metric M computed based on a data collection C : How reliable is this paper? More specifically, (a) What happens if C is replaced with another set of data $C1$? (b) How good is M ?

Indeed, a recommender systems have a variety of properties that may affect user experience, such as accuracy, robustness, scalability, and so forth.

Hence, various parameters must be tuned to generate more accurate predictions. Most of the effort made when developing this work was experimenting novel solutions to upgrade the system performance results in rating prediction and recommendation performers. We present in this part, the protocol process that can be used to evaluate any CARS approach for different businesses.

7.4.1 Datasets

A dataset is a major component consists of a collection of objects related to each other to support the research evaluation [45]. In the world of recommender systems, it is a common practice to use public available datasets from different application environments in order to evaluate and compare the performance of recommendation algorithms.

In general, the evaluation of performance of any context-aware recommendation model is based on four popular contextual real-world datasets from various domains: music, food, and movie. This variation enables us to assess the performance of the proposed models across a range of different datasets, each with different characteristics. We provide in the following more details about subset of these datasets.

- **Music dataset** [5] is collected from a mobile application recommending music tracks to the passengers involved in various driving and traffic conditions. The dataset contains 8 contextual dimensions and 34 contextual conditions in total.
- **Food dataset** [38] represents a contextual food preference dataset collected from a survey containing users ratings on the food menu in the context of different degrees of hunger.
- **Movie dataset** [57] is a context-aware movie dataset collected from surveys. Students were asked to rate movies in different contexts. Three contextual dimensions were captured: Time (weekend, weekday), Location (home, cinema), and Companion (alone, family, partner).
- **LDOS-CoMoDa dataset** is a movie-rating dataset collected by Odic et al. [27]. It contains ratings acquired in contextual situations that are described as a set of different contextual conditions coming from 12 various contextual dimensions, for example, social, day type, location, and mood.

The properties of these datasets are summarized in Table 7.1.

Table 7.1 Description of the used datasets

Dataset	# of users	# of items	# of ratings	# of contextual dimensions	# of contextual conditions
Music	41	139	3940	8	34
Food	212	20	6360	6	8
Movie	97	79	5035	3	12
LDOS-CoMoDa	185	4138	2297	12	49

7.4.2 Evaluation Methodology

To evaluate a CARS, the evaluation methodology defines the followed experimental protocol that can fall into one of the two main levels: the offline or the online evaluations [6, 25].

7.4.2.1 Offline Evaluation

Offline evaluations are popular methods performed in the literature to assess recommendation approaches. This kind of evaluation is realized by using collected datasets of items gathering user interactions. User behavior when interacting with the recommendation system is simulated by using the collected dataset. Since the method deals with the users behavior collected in the past, the offline evaluation does not need any interaction with real users allowing the comparison of wide range of approaches at low cost. However, offline evaluations cannot measure the effect of the recommendation system on the user behavior, they only give a first level performance evaluation by providing a good approximation of how the system would behave with real users. The basic structure for offline evaluation process is based on the train-test and cross-validation techniques. The dataset containing the information of users, items, and ratings is often partitioned. Part of this data is used to infer the optimal utility function and referred to as training set. The other part is known as the testing set and adopted to measure the recommendations performance. When the same data is used for both training and evaluation, the dataset splitting is useful for preventing algorithms from over fitting to the evaluation testing set. To split the dataset, different ways could be adopted, knowing that the chosen manner depends on the domain of application and its constraints [3].

7.4.2.2 Online Evaluation

Online evaluation is generally conducted with real users that interact with the system and give feedback based on their experience. This type of evaluation focuses on measuring the change in user behavior during the interaction with different recommender systems. Questionnaires and user studies are provided to the user

for evaluating the accuracy and performance of the RS. The risk taken when carrying out online evaluation is requiring plenty of efforts in gathering the feedback responses from users. Moreover, comparing several algorithms through online experiments is expensive and time-consuming. Besides choosing an evaluation methodology, evaluation metrics are also necessary to assess the performance of recommender systems. Numerous evaluation metrics have been proposed in the RS and CARS systems literature. However, they are generally based on the famous recall and precision metrics yet used in classical information retrieval.

7.4.3 Evaluation Metrics

We now turn our attention to the different metrics adopted to assess the performance of recommender systems. A distinction needs to be made between the evaluation metrics by taking into account the goal of the system itself. Generally, these metrics can be categorized into *prediction accuracy metrics* that determine how well a system can predict the appropriate rating for an item and *top-N metrics* that measure the suitability of top-N recommendations to users. We present in the following the commonly used evaluation metrics:

7.4.3.1 Prediction Accuracy Metrics

Prediction accuracy is considered as the most discussed property in the recommendation literature. It measures how close the recommendation system rating predictions are to the users real ratings. To date, the majority of RS are based on a rating prediction phase, where the main assumption is that a RS that produces more accurate predicted ratings will be more preferred by the user. This category of evaluation metrics comprises the well known Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) which are considered as standard metrics for many RS such as the Netflix Prize [8]. The lower the error value, the better the predictive accuracy of the recommender system is.

- **Mean Absolute Error (MAE)** measures the average absolute deviation between the system's predicted ratings and the user's actual ratings. It is given by the following equation:

$$\text{MAE} = \frac{1}{N} \sum_{i \in N} |r_{ui} - \hat{r}_{ui}| \quad (7.1)$$

where:

- N : the total number of recommended items.
- \hat{r}_{ui} : the predicted rating of user u for item i .
- r_{ui} : the real rating of user u for item i .

- **Root Mean Squared Error (RMSE)** measures the quadratic error and it is hence more sensitive to large errors, since the errors are squared before they are averaged. This means that the RMSE is useful when large errors are especially undesirable. The RMSE is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i \in N} (r_{ui} - \hat{r}_{ui})^2} \quad (7.2)$$

7.4.3.2 Top-N Metrics

For evaluating the top-N recommendations, the used evaluation metrics focus on measuring the quality of top-N recommendation lists generated by RS. In this family of measures, we found two popular metrics borrowed from the field of information retrieval: *Precision* and *Recall*.

- **Precision@N** measures the fraction of relevant recommended items in the top-N position and is defined as follows:

$$\text{Precision@N} = \sum_{i=1}^N \frac{\text{rel}(i)}{N} \quad (7.3)$$

Here, $\text{rel}(i)$ indicates the relevance level of the item at position i , $\text{rel}(i) = 1$ if the item is relevant and $\text{rel}(i) = 0$ otherwise.

- **Recall@N** calculates the ratio of selected relevant items returned in the top-N position, to the total number of available relevant items Nr . Recall can be computed with the help of the following equation:

$$\text{Recall@N} = \sum_{i=1}^N \frac{\text{rel}(i)}{Nr} \quad (7.4)$$

Increasing the recommendation list size may result in a higher recall but a lower precision, since a longer recommendation list tends to include relevant items. The *F-measure* evaluates the balance between these two metrics and is described as follows:

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7.5)$$

Besides evaluating the relevance of items in the recommendation list, it is also important to evaluate the ranking quality. In particular, we introduce the following two widely used ranking measures *Discounted Cumulative Gain (NDCG)* and the *Mean Reciprocal Rank (MRR)*.

- **NDCG@N** Normalized Discounted Cumulative Gain is calculated based on computing Discounted Cumulative Gain (DCG) which measures the effectiveness of a ranked list based on items relevance. NDCG is the normalized variant of DCG, where Ideal DCG (IDCG) is the best possible DCG.

$$\begin{aligned} \text{DCG@N} &= \frac{1}{N} \sum_{i=1}^N \frac{2^{\text{rel}(i)} - 1}{\log_2(i + 1)} & \text{IDCG@N} &= \frac{1}{N} \sum_{i=1}^k \frac{1}{\log_2(i + 1)} \\ \text{NDCG@N} &= \frac{\text{DCG@N}}{\text{IDCG@N}} \end{aligned} \quad (7.6)$$

- **MRR@N** Mean Reciprocal Rank is described as the multiplicative inverse of the rank of the first relevant item, L represents the relevant items list in the testing set for each user, and Rank_i denotes the position of the relevant item i in the recommendation list.

$$\text{MRR@N} = \frac{1}{|L|} \sum_{i=1}^{|L|} \frac{1}{\text{rank}_i} \quad (7.7)$$

7.4.3.3 Alternative Performance Metrics

While most research in recommender systems has focused on accuracy metrics, additional characteristics of recommendations could be taken into consideration. Thus, other performance metrics such as novelty and diversity may be measured [12]. Novelty and diversity are different though related notions.

- **Novelty** evaluates whether the recommended items are new to the user or not. It would be interesting if the user is recommended with novel items. Novelty can be measured by comparing the top-N recommendations against already used or rated recommendations. Given I_R , the set of items that have been previously recommended to a user u , and I_T , the set of the top-N recommended items to u , novelty for each user u can be defined as follows:

$$\text{Novelty}_u = \frac{|I_T \setminus I_R|}{|I_T|}$$

The average $\frac{1}{N} \sum_{u=1}^N \text{Novelty}_u$ can be interpreted as the measurement of novelty, where N denotes the number of users.

- **Diversity** is related to how dissimilar the recommended items are with respect to each other. The diversity can be determined using the items content (e.g., movie or music genres) or the items ratings by measuring Intra-List Similarity (ILS) [59]. ILS calculates the similarity between two items i_n and i_m in the

recommendation list L using a similarity metric such as Jaccard similarity coefficient [9]. For a user u , ILS can be computed as:

$$ILS_u = \frac{1}{2} \sum_{i_n \in L} \sum_{i_m \in L} sim(i_n, i_m)$$

From here, the overall ILS can be calculated as the average over all users.

7.4.4 *Recommender Systems Platforms*

The wide array of recommendation algorithms proposed over the years brings a challenge in their reproduction and comparison. Therefore, multiple open-source frameworks exist for this purpose. Many implementations of recommender algorithms are available, especially for collaborative filtering algorithms. A lot of tools are free, open-source projects that researchers can use. However, they provide only a few classic recommendation algorithms. The most two relevant ones are LibRec (Library for Recommender system)[21] and CARSKit (Context-aware Recommender system) [57]. LibRec is depicted to baseline and social recommender algorithms, whereas CARSKit uses the implementations for no-contextual recommender algorithms from the LibRec and adds the required functionality to implement contextual recommender systems. In any experiment, the use of any implementation of the recommender algorithm and a series of evaluation metrics as provided by the two latter ones to study of the two main problems of recommender systems, rating prediction and item recommendation implements a suite of state-of-the-art recommendation algorithms as well as the traditional methods. In addition, a series of evaluation metrics are implemented including diversity-based metrics which are rarely enabled in other libraries. LibRec provides a platform for fair comparisons among different algorithms in multiple aspects, given the fact that the evaluative performance depends on data characteristic. It also provides a high flexibility for expansion with new algorithms [18].

7.4.5 *Conventional Methods in Contextual Recommender Systems*

To evaluate any CARS solution, a comparative study must be done with a baseline. In general, the recommendation algorithms can be chosen, for example, from the java based context-aware recommendation engine [57]. A subset of algorithms that can be chosen for comparison are described below:

1. **User-oriented K-Nearest Neighbors (UserKNN)** [53] represents a neighborhood collaborative filtering algorithm on the basis of users similarity.

2. **Item-oriented K-Nearest Neighbors (ItemKNN)** [53] represents a neighborhood collaborative filtering algorithm on the basis of items similarity.
3. **Differential Context Weighting (DCW)** [55] introduces the contextual weighting in the rating prediction process through a weighted similarity measure.
4. **Singular Value Decomposition model based on implicit feedback (SVD++)** [26] represents a matrix factorization model using users history information.
5. **List-Rank Matrix Factorization (LRMF)** [50] refers to a matrix factorization ranking model that joins the list-wise learning with MF.
6. **Context-Aware Matrix Factorization (CAMF)** [5] represents an extended MF model that integrates contextual information in the rating prediction process. We tried its three variants (CAMF-C, CAMF-CI, and CAMF-CU) and we only present the best performing one, denoted by *CAMF-Dev*.
7. **Multidimensional Context Similarity (CAMF-MCS) model** [56] refers to a CAMF algorithm considering the contextual correlation aspect using a multidimensional space.
8. **Fuzzy Weighting Recommender (FWR)** Inspired by the idea of the paper [55], the rating prediction formula of Resnick's algorithm [42] to generate contextual ratings prediction through a novel proposal called Fuzzy Weighting Recommender (FWR) is adopted [18]. In this prediction process, the notion of contextual situations similarity is introduced, where the more close the contextual situations of two ratings were given, the more reliable those ratings for further predictions. Nevertheless, this effect should be restricted since integrating contexts with low similarity can lead to adding noise to the predictions. Thus, a set of similarity thresholds are introduced to filter ratings, for the each component.

According to FWR, the predicted rating $P_{a,i,\sigma}$ that a given user a is expected to attribute to the item i depending to his contextual situation is computed as follows:

$$P_{a,i,\sigma} = \bar{\rho}(a, \sigma_3, \epsilon_3) + \frac{\sum_{n \in N_{a,\sigma_1,\epsilon_1}} (\rho(n, i, \sigma_2, \epsilon_2) - \bar{\rho}(n, \sigma_2, \epsilon_2)) \times sim_w(a, n, \sigma_4, \epsilon_4)}{\sum_{n \in N_{a,\sigma_1,\epsilon_1}} sim_w(a, n, \sigma_4, \epsilon_4)} \quad (7.8)$$

9. **CAMF-MCS strategies** The majority of the surveyed Context-Aware Matrix Factorization algorithms (CAMF) [4, 54], cannot fully capture the impact of the relevant contextual dimensions as well as their associations on the predicted ratings. This proposal consists of two strategies extended from the correlation based CAMF-MCS model suggested in [56]. Both of the two proposed strategies [18], apply a common rating prediction formula (Eq. 7.9) highlighting the notion of “contextual correlation.”

$$\hat{r}_{u,i,s_t} = \mathbf{q}_i \cdot \mathbf{p}_u \cdot Corr(s_t, s_E) \quad (7.9)$$

In the rating formula 7.9, both items and users are characterized by vectors. In fact, each item i is associated with an item vector denoted \mathbf{q}_i and each user u is associated with a user vector denoted \mathbf{p}_u . Those vectors values are the weights on different latent factors. Precisely, the elements in \mathbf{q}_i indicate the extent to which the item i obtains those latent factors. For the vector \mathbf{p}_u , its elements indicate how much users like those latent factors. The function denoted $(Corr(s_t, s_E))$ predicts the correlation or the similarity between a current contextual situation s_t in which the user u consume the item i and an empty contextual situation s_E .

7.5 Experimental Results: Case Study

In this section, we present a subset of experimental studies. Before conducting the experimental evaluation, we begin by performing preliminary experiments by presenting a parameter sensitivity analysis in order to set the optimum values of these parameters to be used for the further evaluation experiments.

The neighborhood-based model (FWR) and CAMF based model (WCAMF-MCS and ICAMF-MCS strategies) are used according to MAE, Precision@N (Prec@N), Recall@N (Rec@N) and NDCG@N with $N \in \{5, 10\}$.

7.5.1 Analyzing Parameter Sensitivity: Impact of the Number of Iterations

We present on this part the adjustment of the number of iterations parameter. We examine the number of iterations required in the Fuzzy Weighting Recommender approach (FWR) and the CAMF-MCS strategies: the weighting strategy (WCAMF-MCS) and the interaction strategy (ICAMF-MCS). Figure 7.2 reports for each dataset the prediction accuracy measured in compliance with the number of iterations.

It is apparent from Fig. 7.2 that on Music and Food datasets, FWR requires 20 iterations to get a peak prediction accuracy. When it comes to the Movie dataset, both methods indicate reduced prediction accuracy when the iterations number goes beyond 60. For the LDOS-CoMoDa dataset, the best performance is achieved by FWR at 50 iterations. For WCAMF-MCS and ICAMF-MCS, we can note that the prediction accuracy is improved when the iterations number reaches 100. We set the suitable iterations number for each method when the best prediction accuracy is achieved.

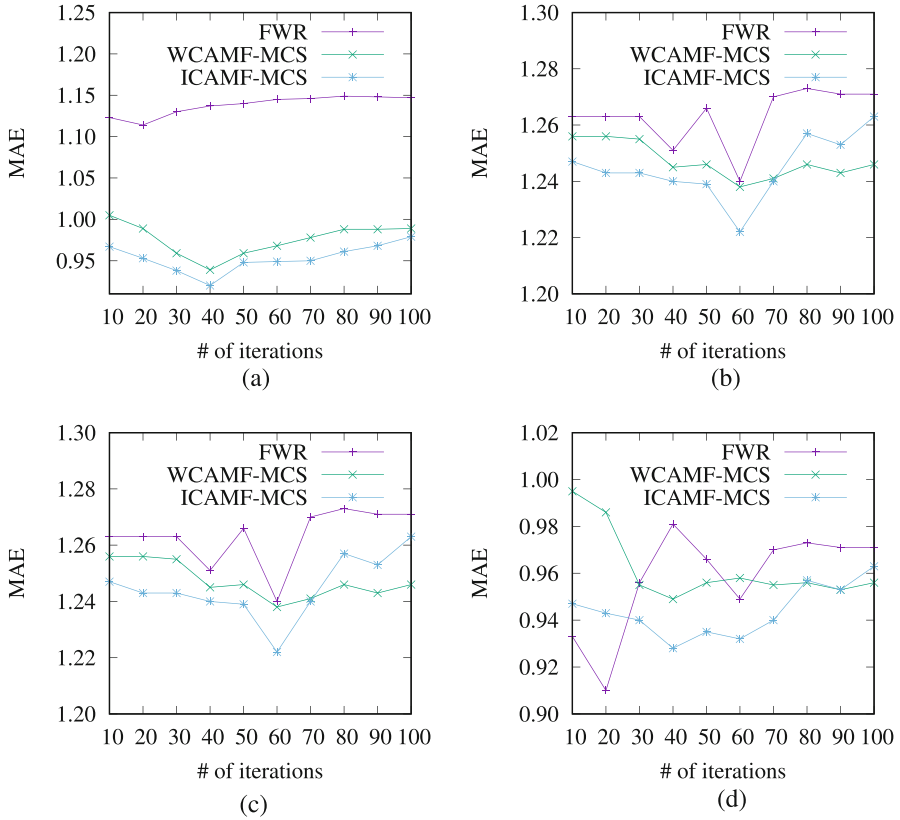


Fig. 7.2 MAE variation in different iterations numbers. (a) Food dataset. (b) Movie dataset. (c) Music dataset. (d) LDOS-CoMoDa dataset

7.5.2 Results

We present in Tables 7.2 and 7.3 the obtained experimental results between some baselines on Music, Movie, LDOS-CoMoDa and Food datasets. We can observe from the two tables below, the CAMF based model is able to outperform the neighborhood-based model. For example, ICAMF-MCS strategy gives an improvement of the Prec@5 value by 28.1%, 16.4%, 45.9%, and 14.5% over FWR, on Music, Movie, LDOS-CoMoDa, and Food datasets, respectively.

Given the fact that the neighborhood-based model can suffer from low accuracy problem due the absence of the knowledge learned about item aspects to produce accurate top-N recommendations. In addition, the neighborhood formation process, especially the user-user similarity computation step requires the calculation of user's interest similarity with all other neighbors to make predictions or recommendations which may increase the computation complexity. However, in the case of having

Table 7.2 Comparison results on the Music and Movie datasets

Dataset	Algorithm	MAE	Prec@5	Prec@10	Rec@5	Rec@10	NDCG@5	NDCG@10
Music	ItemKNN	0.983	0.015	0.014	0.043	0.079	0.040	0.045
	UserKNN	1.087	0.013	0.015	0.038	0.091	0.043	0.042
	DCW	1.064	0.058	0.052	0.090	0.144	0.121	0.123
	FWR	0.911	0.064	0.070	0.106	0.161	0.143	0.148
	SVD++	0.965	0.036	0.025	0.183	0.179	0.117	0.110
	LRMF	1.270	0.024	0.017	0.186	0.134	0.077	0.075
	CAMF-Dev	1.001	0.014	0.018	0.142	0.150	0.042	0.037
	CAMF-MCS	0.998	0.033	0.031	0.118	0.166	0.112	0.092
	WCAMF-MCS	0.939	0.078	0.071	0.191	0.172	0.128	0.129
	ICAMF-MCS	0.920	0.082	0.079	0.198	0.195	0.151	0.141
Movie	ItemKNN	1.229	0.052	0.044	0.263	0.248	0.210	0.231
	UserKNN	1.242	0.055	0.045	0.275	0.285	0.202	0.201
	DCW	1.248	0.046	0.052	0.295	0.302	0.261	0.266
	FWR	1.240	0.061	0.062	0.302	0.322	0.346	0.283
	SVD++	1.688	0.057	0.028	0.268	0.104	0.222	0.105
	LRMF	1.395	0.053	0.042	0.276	0.251	0.224	0.136
	CAMF-Dev	1.229	0.048	0.045	0.281	0.311	0.226	0.119
	CAMF-MCS	1.529	0.052	0.049	0.391	0.351	0.245	0.123
	WCAMF-MCS	1.238	0.069	0.064	0.404	0.396	0.246	0.142
	ICAMF-MCS	1.223	0.071	0.065	0.496	0.403	0.248	0.184

a sufficiently small number of users, neighborhood-based model can outperform matrix factorization based model. For example, FWR improves the best performing strategy of CAMF based model by 5% and 53.8% in terms of NDCG@10 on Music and Movie datasets, respectively. We can observe a little difference between the two strategies ICAMF-MCS and WCAMF-MCS. Most commonly, the ICAMF-MCS strategy gives a better performance than WCAMF-MCS strategy. In this respect, we can note that, ICAMF-MCS slightly improves the MAE value over WCAMF-MCS by 2%, 1.2% and 1.2% on Music, Movie and Food datasets, respectively. ICAMF-MCS strategy is also able to beat WCAMF-MCS strategy in terms of Prec@10 and Rec@10 on LDOS-CoMoDa dataset by an improvement of 23.1% and 80.7%, respectively. The obtained experimental results show the superior performance of the ICAMF-MCS strategy especially on rich contextual datasets. In fact, this latter strategy takes into account the interaction that may exist between the relevant contextual dimensions according to their fuzzy measures. Therefore, the strategy that considers correlated contextual dimensions outperforms the one considering independent contextual dimensions. As a result, the interaction among the relevant contextual dimensions may be considered as a better framework to understand and represent the contextual effects on recommendation. For instance, a user may more precisely decide a movie if the time contextual dimension is correlated with companion dimension rather than considering these contextual dimensions separately.

Table 7.3 Comparison results on the LDOS-CoMoDa and Food datasets

Dataset	Algorithm	MAE	Prec@5	Prec@10	Rec@5	Rec@10	NDCG@5	NDCG@10
LDOS-CoMoDa	ItemKNN	0.973	0.007	0.006	0.024	0.020	0.024	0.030
	UserKNN	0.952	0.004	0.005	0.019	0.025	0.021	0.027
	DCW	0.830	0.002	0.005	0.017	0.026	0.019	0.022
	FWR	0.821	0.037	0.025	0.026	0.028	0.031	0.033
	SVD++	0.871	0.024	0.025	0.018	0.017	0.021	0.040
	LRMF	2.004	0.011	0.012	0.012	0.009	0.007	0.005
	CAMF-Dev	0.867	0.022	0.020	0.027	0.022	0.006	0.006
	CAMF-MCS	1.021	0.042	0.032	0.016	0.010	0.008	0.005
	WCAMF-MCS	0.848	0.048	0.039	0.033	0.026	0.018	0.059
	ICAMF-MCS	0.804	0.054	0.048	0.043	0.047	0.033	0.061
	ItemKNN	1.183	0.060	0.065	0.106	0.146	0.119	0.118
	UserKNN	1.214	0.038	0.046	0.099	0.120	0.118	0.120
	DCW	1.206	0.069	0.032	0.105	0.118	0.101	0.107
	FWR	1.114	0.076	0.040	0.145	0.151	0.125	0.122
SVD++	1.119	0.062	0.050	0.131	0.150	0.123	0.128	
Food	LRMF	1.270	0.065	0.047	0.172	0.147	0.128	0.118
	CAMF-Dev	1.007	0.040	0.048	0.122	0.146	0.138	0.117
	CAMF-MCS	1.529	0.080	0.072	0.144	0.172	0.137	0.123
	WCAMF-MCS	0.938	0.086	0.073	0.190	0.177	0.139	0.143
	ICAMF-MCS	0.927	0.087	0.082	0.199	0.197	0.154	0.134

As expected, Tables 7.2 and 7.3 show that the neighborhood-based CF model (FWR) can significantly improve the rating accuracy metric MAE over the previous popular neighborhood-based CF approaches (ItemKNN, UserKNN and DCW). For example, FWR achieves an MAE value equals to 1.114 while the best performing neighborhood-based baseline achieves an MAE value equals to 1.183 on Food dataset. It also can be found that, on Music dataset, FWR improves the MAE value from 0.983 (the MAE of the best performing neighborhood-based baseline) to 0.911. Furthermore, when it comes to the top-N recommendation task, FWR is also able to achieve higher ranking metric values and thus outperforms the neighborhood-based baselines. For instance, on LDOS-CoMoDa dataset, FWR gives an improvement in terms of Rec@5 by 8.3% over ItemKNN, 36.8% over the UserKNN and 52.9% over the DCW. Therefore, the comparative neighborhood-based CF models always show lower results than the neighborhood based model FWR. A possible explanation for this is that these baselines ruled out the influence of contextual dimensions relevancy and interaction in determining suitable neighbors with similar contexts which may increase the computational complexity in the neighborhood formation process and thus decrease recommendation accuracy.

Regarding the comparison between the matrix factorization-based models, we can notice that CAMF based models (CAMF-Dev and CAMF-MCS) work better than MF models (SVD++ and LRMF). Nevertheless, it can be found that, in terms of MAE, MF models such as SVD++ can improve the CAMF-MCS by 36.6% on Food dataset, this may have occurred due to the small contextual conditions number in this dataset.

The two strategies (WCAMF-MCS and ICAMF-MCS) can achieve a superior recommendation performance over prior CAMF models, particularly ICAMF-MCS strategy. It outperforms Rec@5 by 41.3% and 76.5% relative to CAMF-MCS and CAMF-Dev, respectively, on Movie dataset. Moreover, on LDOS-CoMoDa dataset, ICAMF-MCS makes better Rec@5 value by 59.3% and 168.7% than CAMF-Dev and CAMF-MCS, respectively.

Let us note that ICAMF-MCS usually obtains the preferable results consistently which prove the accuracy of the interaction based CAMF strategy and confirms the efficiency of employing weighted correlated contextual dimensions in the prediction process using factorization techniques.

7.6 CARS Systems on Business

Recommender systems (RSs), initially introduced to address the problem of improving the customer experience and retention in e-commerce sites [30, 44, 51] has since become a ubiquitous and often anticipated functionality of many online interactions, from movie and song recommendations [13, 37, 49] to applications related to tourism [10, 19], social networks [36, 52], health [15], and many more. One of the important potential benefits of recommendation systems is their ability to continuously adapt to the preferences of the user.

The applications of recommender systems include recommending movies, music, television programs, books, documents, websites, conferences, tourism scenic spots and learning materials, and involve the areas of e-commerce, e-learning, e-library, e-government, and e-business services.

Collaborative filtering (CF) is the most popular algorithm used to design various applications and sites for recommender systems such as Facebook, Twitter, Google, LinkedIn and Netflix. For example, Same Mckinsey¹³ study highlights that 75% of Netflix viewing is driven by recommendations. The underlying idea behind CF is that users with common interests in the past are more likely to keep exhibiting similar interests in the future. The principal property to work with collaborative filtering are the ratings given by users for items.

Amazon.com uses item-to-item collaborative filtering recommendations on most pages of their website and e-mail campaigns. According to McKinsey, 35% of Amazon purchases are thanks to recommendations systems. They suggest the most relevant items to buy and, as a result, increase a company's revenue. These suggestions are based on users' behavior and history that contain information on their past preferences.

Spotify generates a new customized playlist for each subscriber called "Discover Weekly" which is a personalized list of 30 songs based on users' unique music taste. A music recommendation engine uses three types of recommendation model: Collaborative Filtering, Natural language processing and Audio file analysis.

Many restaurant recommendation applications are available for public. For example, Google Maps,¹⁴ helps restaurant diners to know what to order. Hence, Maps has transformed from just being a service that offers directions for a commute to more like a search engine for finding coffees, restaurants, shopping centers, etc. Google Maps uses the user's current location as the search query to rank the nearby POIs and then present them to the user. Another popular solution for Restaurant recommendation is the Yelp2app.¹⁵ It provides users with many options, including selecting the price range, sorting restaurants by distance, and many other sophisticated options.

7.7 Conclusion

In this chapter, we introduced an overview of recommender systems and explained how basically these systems work. Therefore, we presented the basic concepts, the recommendation problem formulation, and the main recommendation techniques as well as their principal limitations. We attempted to extend existing knowledge and trace the evolution of the recommendation problem by considering recent emerging

¹³ <https://www.Mckinsey.com/>.

¹⁴ <https://techpp.com/2020/03/09/personalised-restaurant-recommendations-google-maps/>.

¹⁵ <https://www.yelp.com>.

trends. We also gave an overview on performance evaluation methodology. We will focus on future work on multi-criteria decision making for CARS systems and we will discuss the main existing approaches in these areas.

References

1. G. Adomavicius et al., Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.* **23**(1), 103–145 (2005)
2. M. Al-Ghossein, Context-aware recommender systems for real-world applications. Ph.D. Thesis. Université Paris-Saclay (ComUE) (2019)
3. K. Arour, S. Zammali, A. Bouzeghoub, Test-bed building process for context-aware peer-to-peer information retrieval evaluation. *Int. J. Space Based Situated Comput.* **5**(1), 23–38 (2015). <https://doi.org/10.1504/IJSSC.2015.067980>
4. L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, in *Proceedings of the Fifth ACM Conference on Recommender Systems* (2011), pp. 301–304
5. L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, in *Proceedings of the Fifth ACM Conference on Recommender Systems. RecSys '11*. Chicago (2011), pp. 301–304. ISBN:978-1-4503-0683-6
6. J. Beel, S. Langer, A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems, in *Research and Advanced Technology for Digital Libraries*, ed. by S. Kapidakis, C. Mazurek, M. Werla (Springer International Publishing, Cham, 2015), pp. 153–168
7. R. Bell, Y. Koren, C. Volinsky, Modeling relationships at multiple scales to improve accuracy of large recommender systems, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2007), pp. 95–104
8. J. Bennett, S. Lanning, N. Netflix, The Netflix Prize, in *In KDD Cup and Workshop in Conjunction with KDD* (2007)
9. E. Blanchard, M. Harzallah, P. Kuntz, A generic framework for comparing semantic similarities on a subsumption hier archy, in *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence* (IOS Press, Amsterdam, 2008), pp. 20–24. ISBN: 978-1-58603-891-5. <http://dl.acm.org/citation.cfm?id=1567281.1567291>
10. M. Braunhofer, F. Ricci, Selective contextual information acquisition in travel recommender systems. *Inf. Technol. Tour.* **17**(1), 5–29 (2017)
11. R. Burke, Hybrid recommender systems: survey and experiments. *User Model. User-Adap. Inter.* **12**(4), 331–370 (2002)
12. P. Castells, N.J. Hurley, S. Vargas, *Novelty and Diversity in Recommender Systems*. Recommender Systems Handbook (Springer, Boston, 2015)
13. K. Chappannarungsri, S. Maneeroj, Combining multiple criteria and multidimension for movie recommender system, in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1 (2009)
14. G. Chen, D. Kotz, A survey of context-aware mobile computing research. in *Dartmouth Computer Science Technical Report TR2000-381* (2000)
15. A. Civit-Balcells, L. Fernandez-Luque, F. Luna-Perejon, H. de Vries, Analyzing, analyzing recommender systems for health promotion using a multidisciplinary taxonomy: a scoping review. *Int. J. Med. Inform.* **1** (2017). <https://doi.org/10.1016/j.ijmedinf.2017.12.018>
16. A.K. Dey, G.D. Abowd, D. Salber, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum. Comput. Interact.* **16**(2–4), 97–166 (2001)

17. R. Dridi et al., An improved context-aware matrix factorization model incorporating fuzzy measures, in *2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2018, Rio de Janeiro, July 8–13, 2018* (IEEE, Piscataway, 2018), pp. 1–8. <https://doi.org/10.1109/FUZZ-IEEE.2018.8491439>
18. R. Dridi et al., Effective rating prediction based on selective contextual information. *Inf. Sci.* **510**, 218–242 (2020)
19. D. Gavalas et al., Mobile recommender systems in tourism. *J. Netw. Comput. Appl.* **39**, 319–333 (2014)
20. D. Goldberg et al., Using collaborative filtering to weave an information tapestry. *Commun. ACM* **35**(12), 61–70 (1992)
21. G. Guo et al., LibRec: a Java library for recommender systems, in *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, June 29–July 3, 2015*, ed. by A.I. Cristea et al., vol. 1388. CEUR Workshop Proceedings. CEUR-WS.org (2015). http://ceur-ws.org/Vol-1388/demo_paper1.pdf
22. L. Hong et al., Context-aware recommendation using role-based trust network. *ACM Trans. Knowl. Discov. Data* **10**(2), 1–25 (2015)
23. Jhamb, Yogesh, “Machine Learning Models for Context-Aware Recommender Systems” (2018). Engineering Ph.D. Theses. 15. https://scholarcommons.scu.edu/eng_phd_theses/15
24. M. Jin et al., Combining deep learning and topic modeling for review understanding in context-aware recommendation, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long Papers) (2018), pp. 1605–1614
25. Y. Kim, S.B. Cho, A recommendation agent for mobile phone users using bayesian behavior prediction, in *Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Sliema (2009), pp. 283–288
26. Y. Koren, Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* **4**(1), 1:1–1:24 (2010). ISSN: 1556-4681
27. A. Košir et al., Database for contextual personalization. *Elek Trotehniski Vestnik* **78**(5), 270–274 (2011)
28. B. Lamche et al., Context-aware recommendations for mobile shopping, in *Proceedings of the Workshop on Location-Aware Recommendations, LocalRec, Co-located with the 9th ACM Conference on Recommender Systems (RecSys), Vienna, September 19* (2015), pp. 21–27
29. Q.-H. Le et al., A state-of-the-art survey on context-aware recommender systems and applications. *Int. J. Knowl. Syst. Sci.* **12**(3), 1–20 (2021)
30. Q. Li, C. Wang, G. Geng, Improving personalized services in mobile commerce by a novel multicriteria rating approach. in *Proceedings of the 17th International Conference on World Wide Web* (2008), pp. 1235–1236
31. A. Livne et al., Deep context-aware recommender system utilizing sequential latent context (2019). Preprint. arXiv:1909.03999
32. P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in *Recommender Systems Handbook* (Springer, Berlin, 2011), pp. 73–105
33. Z. Meng et al., Variational bayesian context-aware representation for grocery recommendation (2019). Preprint. arXiv:1909.07705
34. F. Meyer, Recommender systems in industrial contexts (2012). Preprint. arXiv:1203.4487
35. S. Michael, W. Peter, *War and Society in Twentieth-Century France*, ed. by M. Scriven, P. Wagstaff (Martin’s Press, New York, 1991), xii, 304 p., [4] p. of plates. ISBN: 0854962921
36. M. Nilashi et al., Analysis of travellers’ online reviews in social networking sites using fuzzy logic approach. *Int. J. Fuzzy Syst.* **21**(5), 1367–1378 (2019)
37. A. Odić et al., A.: relevant context in a movie recommender system: users’ opinion vs. statistical detection, in *Proceedings of the 4th Workshop on Context-Aware Recommender Systems, CARS, September 9, Dublin* (2012)
38. C. Ono et al., *Context-Aware Preference Model Based on a Study of Difference Between Real and Supposed Situation Data* (Springer, Berlin, 2009), pp. 102–113

39. A.M. Otebolaku, M.T. Andrade, Context-aware personalization using neighborhood-based context similarity, in *Wireless Personal Communications* (2016), pp. 1–24
40. N. Polatidis, C.K. Georgiadis, Mobile recommender systems: an overview of technologies and challenges, in *2013 Second International Conference on Informatics & Applications (ICIA)* (IEEE, Piscataway, 2013), pp. 282–287
41. P. Resnick, H.R. Varian, Recommender systems. *Commun. ACM* **40**(3), 56–58 (1997)
42. P. Resnick et al., GroupLens: an open architecture for collaborative filtering of netnews, in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (1994), pp. 175–186
43. F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in *Recommender Systems Handbook* (Springer, Berlin, 2011), pp. 1–35
44. C. Richthammer, G. Pernul, Situation awareness for recommender systems. *Electron. Commer. Res.*, 1–24 (2018)
45. K.V. Rodpysh, S.J. Mirabedini, T. Banirostam, Correction to: employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems. *Electron. Commer. Res.* **22**(1), 223 (2022)
46. I.H. Sarker, A machine learning based robust prediction model for real-life mobile phone data. *Internet Things* **5**, 180–193 (2019)
47. I.H. Sarker, Y.B. Abushark, A.I. Khan, ContextPCA: predicting context-aware smartphone apps usage based on machine learning techniques. *Symmetry* **12**(4), 499 (2020)
48. B.N. Schilit, M.M. Theimer, Disseminating active map information to mobile hosts. *IEEE Netw.* **8**(5), 22–32 (1994). ISSN: 0890-8044. <https://doi.org/10.1109/65.313011>
49. A. Sen, M. Larson, From sensors to songs: a learning-free novel music recommendation system using contextual sensor data, in *Proceedings of the Workshop on Location-Aware Recommendations, Local-Rec, Vienna, September 19* (2015), pp. 40–43
50. Y. Shi, M.A. Larson, A. Hanjalic, List-wise learning to rank with matrix factorization for collaborative filtering, in *Proceedings of the Fourth ACM Conference on Recommender Systems* (2010), pp. 269–272. ISBN: 978-1-60558-906-0
51. K. Shin et al., One4all user representation for recommender systems in E-commerce. *CoRR* abs/2106.00573 (2021). arXiv: 2106.00573. <https://arxiv.org/abs/2106.00573>
52. T. Stepan et al., Incorporating spatial, temporal, and social context in recommendations for location-based social networks. *IEEE Trans. Comput. Soc. Syst.* **3**(4), 164–175 (2016)
53. H.-M. Wang, G. Yu, Personalized recommendation system K neighbor algorithm optimization, in *Proceedings of International Conference on Information Technologies in Education and Learning (ICITEL 2015)*, Atlantis Press. pp. 1–4
54. Y. Zheng, Context-aware collaborative filtering using context similarity: an empirical comparison. *Information* **13**(1), 42 (2022)
55. Y. Zheng, R. Burke, B. Mobasher, Recommendation with differential context weighting, in *21th International Conference, Rome, June 10–14* (2013)
56. Y. Zheng, B. Mobasher, R. Burke, Incorporating context correlation into context-aware matrix factorization, in *Proceedings of the International Conference on Constraints and Preferences for Configuration and Recommendation and Intelligent Techniques for Web Personalization*, Buenos Aires (2015), pp. 21–27
57. Y. Zheng, B. Mobasher, R.D. Burke, CARSKit: a Java-based context-aware recommendation engine, in *IEEE International Conference on Data Mining Workshop, ICDMW, Atlantic City, November 14–17* (2015), pp. 1668–1671
58. X. Zheng et al., A new recommender system using context clustering based on matrix factorization techniques. *Chinese J. Electron.* **25**(2), 334–340 (2016)
59. C.-N. Ziegler et al., Improving recommendation lists through topic diversification, in *Proceedings of the 14th International Conference on World Wide Web, WWW '05, Chiba* (ACM, New York, 2005), pp. 22–32. ISBN: 1-59593-046-9. <https://doi.org/10.1145/1060745.1060754>. <http://doi.acm.org/10.1145/1060745.1060754>