Expert Systems With Applications 183 (2021) 115375

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# MORec: At the crossroads of context-aware and multi-criteria decision making for online music recommendation

Imen Ben Sassi [1,*], Sadok Ben Yahia, Innar Liiv

*Tallinn University of Technology, Akadeemia tee 15a, Tallinn 12618, Estonia*

ARTICLE INFO

*Keywords:*
Recommender systems
User-based Study
Multi-criteria recommendation
Context aware recommender system (CARS)
Clustering
Music online recommendation (MORec)

ABSTRACT

Context-aware recommender systems have received considerable attention from industry and academic areas. In this paper, we pay heed to the growing interest in integrating context-awareness and multi-criteria decision making in recommender systems, to deal with the most pressing challenges in music recommender systems, namely the diversity of the recommended playlist, the scalability of the system, and the cold start problem. This paper introduces a new multi-criteria recommendation approach, named MORec, which generates Top-N music recommendations by bootstrapping the system using beforehand collected data. We usher by gauging the relevance of contextual information from the relation between three elements: user, music genre, and the user's context. Then, we apply an aggregation technique to uncover the relationship between the context and the overall rating. Besides, we apply the K-means algorithm to generate a predictive model that comprises clusters of similar contexts defining the association between contextual dimensions and music genres. Carried out experiments emphasize very promising results of our approach in terms of clustering quality, compared to the Partitioning Around Medoids algorithm in terms of connectivity and stability. The comparison versus pioneering recommendation baselines underscored the effectiveness of MORec in terms of recommendation quality and usefulness.

## 1. Introduction

Recommender systems (RS) recommend items of interest to users, which are usually selected based on their online behavior history, such as comments, shares, downloads, clicks, and likes, or their social relationships, like friends and similar users. Over the past decade, several categories of RS have been developed (Ricci, Rokach, Shapira, & Kantor, 2010; Ricci, Rokach, & Shapira, 2015). Traditional ones assume that there is a set of Users $U$, and a set of Items $I$ that can be recommended to them. The utility function $R$, that measures the relevance of an item $i \in I$ to a user $u \in U$, is defined as a mapping $R : U \times I \longrightarrow R_0$, where $R_0$ represents the overall rating given by a user on one item. In traditional RS, $R(u, i)$ defines a single criterion value. However, there are some recent works, which consider this assumption as insufficient. Indeed, the relevance of an item to a particular user may vary depending on the utility-related aspect considered when evaluating the recommendation. Sometimes, adding additional criteria can alter the users' choices and provide better recommendations. For instance, Yelp[2] application uses several criteria, like price, category, companion, location, and time, to suggest to people suitable local businesses. In Yahoo! Movies,[3] users can rate a given movie based on its story, action, direction, visual effects, besides its overall rating (Adomavicius, Manouselis, & Kwon, 2011). Hence, incorporating multiple criteria ratings may help to enhance the quality of the recommendation with the new complex representation of users' preferences. Multi-criteria recommender systems (MCRS) are RS that use users' multiple ratings of items regarding various aspects (Adomavicius et al., 2011; Adomavicius & Kwon, 2015; Zheng, 2017), in order to anticipate their preferences. Therefore, the utility function is no longer a function with a single rating (the overall rating) but uses multi-criteria ratings, which represent more complex user preferences. This function is represented as $R : U \times I \longrightarrow R_1 \times R_2 \times \ldots \times R_k$, where $R = (R_1, R_2, \ldots, R_k)$ specifies ratings corresponding to each aspect of the set of criteria $Cr = (Cr_1, Cr_2, \ldots, Cr_k)$ (Adomavicius et al., 2011).

Context-aware recommender systems (CARS) represent another

* Corresponding author.
*E-mail addresses:* imen.ben@taltech.ee (I. Ben Sassi), sadok.ben@taltech.ee (S. Ben Yahia), innar.liiv@taltech.ee (I. Liiv).
[1] ORCID ID: https://orcid.org/0000-0001-8772-9731
[2] https://www.yelp.com.
[3] https://www.yahoo.com/entertainment/movies.

https://doi.org/10.1016/j.eswa.2021.115375
Received 14 July 2020; Received in revised form 16 April 2021; Accepted 6 June 2021
Available online 16 June 2021

category of RS which aims to provide pertinent recommendations based on users' contextual situations. The major challenge of CARS is to come up with a flexible and exhaustive consideration of the context to adapt their recommendations to the users' current contexts. We can see context as any information which can characterize the situation of a person and can be considered relevant to the interaction between her and an application, including the user and application themselves (Dey & Abowd, 1999). The idea behind CARS is that interests are multiple, heterogeneous, changing, and even contradictory, and can be influenced by multiple factors, called context. For example, people usually prefer rhythmic music when working out and like quiet music when studying.

In a music scenario, multiple reasons led us to consider the music recommendation as an extremely complex task (Kaminskas & Ricci, 2012; Schedl, Zamani, Chen, Deldjoo, & Elahi, 2018). On one hand, the nature of music content is very challenging to be modeled, analyzed and classified based on audio attributes, like timbre, melody, and rhythm, as well as musical metadata, like album, artist, song, etc. On the other hand, human musical interests cannot be straightforwardly understood. Indeed, a wide variety of factors may influence a person's musical preferences, like age, gender, personality, musical education, origin, occupation, and socio-economic background (Skowron, Lemmerich, Ferwerda, & Schedl, 2017). As a result, it is difficult for systems to decide whether a particular user can like or dislike a piece of music. In order to simplify effective music filtering, music RS has received great attention from both the industry and academia area (Schedl et al., 2018). For example, Sourcetone[4] allows users to listen to music based on how they want to feel while accomplishing their activities, in order to enhance their mood, activity, and health. The online App uses sliders to enable users to select the timeline, songs will be selected from and the level of songs positivity and excitement. In the same context, Last.fm[5] makes use of time and the user's location to offer him/her the upcoming events and top songs in his/her country. Musicovery[6] company integrates subjective elements into its music playlist recommendation system by using emotional and contextual tags attributed by experts to help people discover music. The used tags are related to both tracks, like period/year, mood (calm, happy, etc.), and activity (driving, working, etc.), and artists, like role and geographic location. In the literature, there have been several cases and studies relying on the multi-criteria decision making (MCDM) techniques to solve recommendation problems (Adomavicius & Kwon, 2015) and others having to make use of context in the recommendation process (Ben Sassi, Ben Yahia, & Mellouli, 2017a). However, to the best of the authors' knowledge, there is no previously published research, in the musical domain which integrates contextual information into the definition of the criteria, generally restricted to item attributes. In this paper, we describe our approach of integrating context in a music RS based on a MCDM. We focus on music playlist recommendations based on a list of ranked music genres, which fits within the user's context. The inceptive idea of our system is to generate recommendations without explicit information about the target user preferences by bootstrapping the system by beforehand collected data. Our solution is based on the matching between the user's current context and the music genre clusters that gather similar contextual dimensions.

Using context in music RS is not new and literature has reported several cases and studies (Ben Sassi et al., 2017a). Notwithstanding, our approach is different from previous efforts since we integrate context-awareness and MCDM to access its effectiveness for music recommendations. In the remainder, we focus on some compelling challenges in music RS, namely *the cold start problem*, *the diversity of the recommended playlist*, and *the scalability problem*. Our research contributions are summarized as follows:

1. We gauge the relevance of the contextual information before the design of our CARS to find out the factors that can influence users' musical preferences.
2. We make the first attempt to integrate context-awareness and MCDM in music RS.
3. We compare our recommendation approach with both traditional baselines and context-aware models and show the effectiveness of integrating context awareness and MCDM to improve music recommendations.

We organize the remainder of the paper as follows. Section 2 reviews music RS and the existing works in the fields of context-aware and multi-criteria recommendations. We discuss in Section 3 the importance of our proposal compared to the state of this art approaches and its utility in real life situations. We highlight, in Section 4, the current challenges in the music RS research area. Before proposing our recommendation approach, we briefly describe, in Section 5, the preliminary study that we conducted in (Ben Sassi, Ben Yahia, & Mellouli, 2017b; Ben Sassi & Ben Yahia, 2021) to find out the contextual information which is important for users regarding music recommendations. Thereafter, in Section 6, we thoroughly describe our music multi-criteria context-aware recommender system (MC-CARS) named MORec. Next, Section 7, shows the experimental results of the introduced approach in terms of clustering and recommendation quality. Finally, Section 8 summarizes our work with a discussion of its limitations and our plans for future research directions.

## 2. Related work

In this paper, we introduce a new context-aware framework for music RS. Our work is related to existing context-aware recommendations and multi-criteria recommendations. In this section, we scrutinize related work in these fields in order to highlight the main similarities and differences between our work and previous ones.

### 2.1. Traditional music recommendations

Traditional music Recommendation Systems are usually divided into three types, namely content-based, collaborative filtering, and hybrid approaches. The authors in (Cheng, Shen, Zhu, Kankanhalli, & Nie, 2017) have studied the effect of music play sequence on the conception of music RS. Thus, they have computed the similarity between songs, called song2vec, by applying word embedding techniques in music play sequences. Next, the similarity of the songs is embedded in matrix factorization and used to propose a k-nearest song regularized matrix factorization model. Knowledge graphs have been considered as useful tools for recommending items. More precisely, the nature of graph information promotes the linkage to other graphs and then the incorporation of additional content. Furthermore, graphs can provide additional connections between items and users, which can be discovered to compute the recommendation list. A knowledge graph has been used, by (Oramas, Ostuni, Noia, Serra, & Sciascio, 2016), to provide information to a hybrid recommendation system based on a collection of documents describing music and sound items. The authors were interested in two recommendation tasks, namely song recommendation problem, which gathers song and artist recommendations for music consumers, and sound recommendation for music producers in online sound sharing platforms. Thus, they have enriched item descriptions and tags with data extracted from two additional knowledge graphs, i.e., DBpedia and WordNet. The final recommendations were computed by combining semantic content-based features inferred from the constructed knowledge graph and collaborative features extracted from the implicit user feedback. The authors in (Kim, Won, Liem, & Hanjalic, 2018) have proposed a hybrid Neural Collaborative Filtering for music playlist RS, which applies a Recurrent Neural Network to make use of textual information from the playlist title. Thus, the playlist title is exploited to

[4] http://www.sourcetone.com.
[5] https://www.last.fm.
[6] http://b2b.musicovery.com.

transform the playlist, based on the character-level N-grams technique, into a latent factor vector representation used for tracks recommendation. Then, a Recurrent Neural Network is combined with a Long Short-Term Memory to encode the N-gram feature. Clustering techniques have also been initially used in collaborative filtering, which represents the most recurrent approach in the music recommendation area. A modified DBSCAN algorithm has been used, in (Kuzelewska & Wichowski, 2015), to generate clusters of similar items given as input for a collaborative filtering item-based RS. The proposed approach has shown its efficiency for the scalability problem since it searches for similarities of the active users' preferred tracks only into the clusters they belong to. A tag-aware dynamic music recommendation framework has been proposed by the authors in (Zheng, Kondo, Zilora, & Yu, 2018). The proposed framework addresses the data sparsity issue by using the music tracks' semantic tags to complement a highly sparse user-item interaction matrix. The authors have combined low-rank matrix factorization with a linear-Gaussian state-space model to discover the temporal evolution of users' music preferences over time. The problem of gray-sheep users, defined by users having unusual preferences that can negatively affect the recommendations, has been studied in (Sánchez-Moreno, Gil González, Muñoz Vicente, López Batista, & Moreno García, 2016). The proposed solution is based on the determination of a coefficient representing the degree to which every user is a gray sheep. More precisely, gray sheep users are identified by considering the artists they listen to and not the ratings they provide. In this way, gray sheep users are those who mostly play unpopular artists. To sum up, traditional music RS only explores the relationship between users and items and content description without considering the influence of contextual factors on users' preferences in the mobile environment.

### 2.2. Context-aware recommendations

In this paper, we pay close attention to the dynamic nature of users' preferences, in different situations, which is strongly related to former CARS studies. Indeed, in the last decade, introducing contextual information, besides those about users and items, has grasped attention in the RS research area (Adomavicius & Tuzhilin, 2011). The context describes the dynamic situation of users. Generative probabilistic models, based on Latent Dirichlet Allocation (LDA), have been used in recommendation problems since their effectiveness to join modeling of users, items, and the meta-data associated with contexts based on latent topics (Srivastava, Hingmire, Palshikar, Chaurasia, & Dixit, 2016). In (Cheng & Shen, 2016), the authors have proposed a smart music RS called VenueMusic, aiming to generate a playlist matching a target venue. The system matches songs and venues based on their semantic features. A new location-aware topic model has been developed aiming to model the associations between the music content and venues in a latent semantic space. Alike to the standard LDA, common features of songs suitable for a venue type are mined in a latent semantic space. Then, songs and venue types are represented in the shared latent space. However, other research works have used deep learning techniques to generate recommendations. The authors, in (Zheng & Jose, 2019), have proposed a new RS, which estimates the user preferences by sequential predictions based on the sequence of context dimensions. They proposed two models based on numerical rates and binary rates. Similarly, Bai et al., in (Bai & Kawagoe, 2018), have combined several contextual information, namely, heart rate, physical condition, and the elapsed time of the user's activity, to propose a background music recommendation method. The proposed approach assumes that music recommendations are in close connection with users' level of fatigue as well as mood and heart rate changes, related to the elapsed time of his/her current activity. In (Abdul, Chen, Liao, & Chang, 2018), the authors have proposed an emotion-aware music recommendation system based on the correlation between the user data; gathering his/her information and music listening, and the music. Thus, they applied the deep convolutional neural networks to extract the latent features from music data for classification. They also used the term-frequency and inverse document frequency (TF-IDF) approach to generate the implicit user ratings for the music. The authors, in (Zangerle, Pichl, & Schedl, 2020), have considered acoustic song features and culture-related features as contextual information used to define a music RS. The proposed approach, based on the learning task for rating prediction, applies Gradient Boosting Decision Trees. Indeed, the authors applied the XGBoost system, which sets the learning objective to logistic regression for binary classification (Chen & Guestrin, 2016). The authors have chosen a classification-based recommendation approach since they focused on user modeling aspects to understand the contribution of individual features of the user model. The authors, in (Wang, Ma, Jiang, Ye, & Zhang, 2020), have proposed a network-based music RS that generates recommendations by extracting topics from textual information. More precisely, the proposed approach extracts textual information from the user's device and then generates the music topic suitable for the current user by applying the LDA technique. Next, they generate recommendations from the song heterogeneous information network. The authors, in (Katarya & Verma, 2018), have constructed a multi-layer context graph with implicit feedback data for music RS, which contains three layers, i.e., user-context layer, item-context layer, and decision context layer. The proposed graph models the interactions between users and items in the corresponding decision context. The authors have applied the depth-first-search Bellman-Ford algorithm to traverse the constructed graph to generate a list of ranked recommendations, as well as a particle swarm optimization to optimize the recommendation results. Users' historical listening records have also been leveraged in (Wang, Deng, Zhang, & Xu, 2018) to propose a context-aware music recommendation approach. More precisely, the authors started by learning the embeddings of music tracks from users' listening records. Next, the learned embeddings are used to model the user's global and contextual preferences as well as active interaction sessions. Finally, the system recommends music tracks that would match the user's global and contextual preferences. The authors in (Véras, Prudêncio, & Ferraz, 2019) have proposed a new RS named CD-CARS for cross-domain context-aware recommendations. The proposed system aims to recommend items from the target domain by discovering the similarities between users based on their ratings and their contexts in different domains. Three contextual dimensions including time, location, and companion in three different domains, i.e., movie, book, and music were combined in the CD-CARS.

Some researches were restricted to a particular dimension of context. The authors, in (Sánchez-Moreno, Zheng, & Moreno-García, 2018), have used temporal information to improve music recommendations by considering the evolution of user preferences over time. More precisely, they simulated users' ratings by aggregating implicit feedback with time dynamics. The authors, in (Volokhin & Agichtein, 2018), have studied the users' intentions for listening to music and their relation with daily activities. Thus, the authors have surveyed to understand the correlation between 8 common activities; including cleaning, commuting, cooking, driving, eating, exercising, shopping, showering, studying, walking, and working; and their corresponding intents. The new proposed intent-aware contextual music recommendation has shown promising results in improving the recommendation quality compared to the Spotify playlist generator based on activity only. The authors in (Andjelkovic, Parra, & O'Donovan, 2019) have developed a new system, named MoodPlay, which takes advantage of musical mood dimensions to allow users to discover music collections. This system is based on Geneva Emotional Music Scales (GEMS) [7] in lieu of the well-used traditional Circumplex model. Users' emotions at different granularity levels were extracted from microblogs in (Deng, Wang, Li, & Xu, 2015) and used to improve music recommendations. The obtained results show that the recommendation quality has been improved by incorporating fine-

[7] GEMS is a hierarchical music-specific model of affect, with three root emotions; namely, Sublimity, Vitality, and Unease.

grained emotion of proper time window. Ayata et al., in (Ayata, Yaslan, & Kamasak, 2018), have learned users' emotions from the signals obtained through wearable physiological sensors, namely galvanic skin response (GSR) and photo plethysmography (PPG) physiological sensors, to propose an emotion-based music recommendation framework. The authors have compared three classification algorithms, including random forest, k-nearest neighbors, and decision tree, to select the best technique for GSR and PPG signals fusion. The authors have underscored the existence of a relationship between GSR and PPG signals and emotional arousal and valence dimensions. Recently, a new emotion-aware computational model, based on affective user profiles, has been proposed by the authors in (Polignano, Narducci, de Gemmis, & Semeraro, 2021). Affective information is collected from social media footprints, i.e., social network messages, and modeled as a vector-based on Ekman's model (Ekman, 1999). The model determines whether a given item is suitable for the current affective state of the user by computing an affective coherence score, which takes into consideration the affective user profile and not-affective item features.

### 2.3. Multi-criteria recommendations

In general, the challenge of MCDM problems is to aggregate attributes of alternatives to a final assessment. In the recommendation domain, MCRS consider multi-criteria ratings to evaluate items based on multiple perspectives. Loosely speaking, MCRS provide additional information by admitting that the relevance of an item for a particular user relies on multiple utility-related criterion (Jannach, Karakaya, & Gedikli, 2012). These latter are usually defined based on item attributes. For example, in (Adomavicius et al., 2011), the movie story, action, direction, and visual effects describe the criteria for a movie RS. Recently, the authors in (Hong & Jung, 2021) have proposed a multi-criteria tensor model for tourism RS that considers both multi-criteria ratings (regarding the food, the service, and the price) and cultural groups. In the same context of Point of Interest (POI) recommendations, the authors in (Zhang, Liu, Wang, & Li, 2021) have defined a MCRS which considers social relationships and criteria preferences. The relationship between the overall rating and criteria ratings is computed by clustering users based on their criteria preferences information. Next, an aggregate function based on the Support Vector Regression (SVR) is trained for each cluster to measure the relationships between the overall rating and criteria ratings regarding 6 dimensions, i.e., location, rooms, services, sleep quality, value, and cleanliness.

### 2.4. Discussion

Interestingly enough, context-aware recommendations can be regarded as an MCDM problem. In other words, it seems reasonable to combine recommendations based on MCDM methods and context by defining MCDM criteria using contextual information such as activities or emotions. Recently, this idea has been studied in (Zheng, Shekhar, Jose, & Rai, 2019), where the authors have exploited several methods to integrate context-aware recommendations and MCDM in the context of educational learning RS. The authors have shown, through their experimental results, that the integration of the two recommendation strategies can be of benefit to more accurate recommendations. The authors in (Zheng, 2017) have also explored the idea to treat criteria preferences as contexts and found that it can improve the performance of recommendation models, whether they are carefully selected.

## 3. Methodological and practical contributions

In order to make the article useful for researchers and practitioners, in this section we compare the music recommendation approaches mentioned above and we mention an example of the use of our method in real-life situations. Tables A.8 and A.9, in Appendix A, sketch the surveyed approaches related to the music recommendation area using the following criteria:

- **Category**: defines the family of the RS.
- **Goal**: presents the major purpose of the RS.
- **Technique**: refers to the of use technique to generate recommendations.
- **Recommended Items**: shows the type of the recommendation, i.e., track or playlist containing a sequence of tracks (audio recordings) (Bonnin & Jannach, 2014).
- **Context Information**: presents the type of the used contextual information. Context may include location, time, or related to the listener's activity.
- **Context Relevance**: in the case when contextual information is used, it defines whether the approach has studied their importance.
- **Beyond Accuracy**: shows the beyond-accuracy measures computed to evaluate the RS.
- **Cold Start**: mentions if there is any solution proposed by the RS whenever it does not have sufficient data associated with the new items/users.

As depicted in Tables A.8 and A.9, many music RSs have integrated contextual factors to improve their recommendation quality. However, only a few approaches have studied the importance of these external factors and their effect on the recommendation. Only the authors in (Volokhin & Agichtein, 2018) have conducted a survey to understand what are the users' intents for listening to music and how they relate to their daily activities (Volokhin & Agichtein, 2018). The results of this survey have been used to improve and evaluate their context-aware music RS. It is worth mentioning that some of the surveyed works have evaluated the effect of context information on recommendation results. For example, in (Deng et al., 2015), the authors have varied the granularity of the emotional context, i.e., 2d-, 7d-, and 21d-emotion, as well as its time window size, i.e., short-term and long-term emotion, to study its effect on recommendation quality.

In addition, information gain has been computed in (Véras et al., 2019; Zangerle et al., 2020) to evaluate the contribution of each contextual attribute to the final model and select only the most relevant contextual attributes of each contextual dimension. Most of the surveyed music RSs have recommended a set of songs. We have organized the recommended items as playlists described by a title like in (Andjelkovic et al., 2019; Kim et al., 2018; Volokhin & Agichtein, 2018; Wang et al., 2020) or as a list of top-n items ordered by their importance (Sánchez-Moreno et al., 2016; Zangerle et al., 2020). We also noticed that only two studies, i.e., (Oramas et al., 2016; Andjelkovic et al., 2019), have evaluated their RS based on beyond accuracy evaluation measures. However, the rest of the studies have limited the evaluation to quantitative measures and failed to include user-centric evaluations. In particular, the authors in (Oramas et al., 2016) have evaluated the capacity of their knowledge graph-based RS to suggest items that users would not readily discover for themselves by generating novel and unexpected results. For that, the authors have computed the Entropy-based Novelty. They have also evaluated the aggregate diversity of their RS, which measures the level of personalization provided by a RS (Adomavicius & Kwon, 2012). Otherwise, the diversity has been evaluated in terms of artists in (Andjelkovic et al., 2019) by comparing the number of unique artists rated and played per user in different conditions. In addition, they have analyzed the user's interaction with the interface in different conditions to understand the impact of the design decisions on users' behavior and measure the RS level of interaction, explanation, and control. And in terms of the cold start problem, some solutions have been proposed by the surveyed papers. First, for the traditional RS category, the authors in (Kim et al., 2018) have represented this problem by playlists with no seed tracks and solved it using a text encoder that exploits the title of the playlist. In (Véras et al., 2019) the addition of user ratings from an auxiliary domain has been employed to overcome the cold start problem. The authors in (Polignano et al., 2021) have used data from Social

**Table 1**
Contextual information and their basic properties.

| Dimension | Attribute | Number of categorical classes | Dimension possible values |
|---|---|---|---|
| Temporal information | Part of the day (Adomavicius et al., 2005) | 3 | morning, afternoon, night |
| | Day of the week (Adomavicius et al., 2005) | 2 | work-day, weekend/day-off |
| Location information | Type of location (Hasan et al., 2013) | 6 | home, work/school, eating, entertainment, recreation, shopping |
| Physical information | Weather (Braunhofer et al., 2013) | 4 | sunny, clear-sky, cloudy, hot, rainy, thunderstorm, snowing |
| Activity information | Activity of daily living (Jiang et al., 2012) | 7 | housework, reflection, sports, transportation, shopping, entertainment, relaxation |
| Emotional information | Emotions (Ekman, 1999) | 6 | joy, sadness, anger, fear, disgust, surprise |
| Social information | Companion (Adomavicius et al., 2005) | 5 | alone, with friends/colleagues, with children, with girlfriend/boyfriend, with family |

**Table 2**
Music genres.

| |
|---|
| Blues, Children's music, Classical, Country, Electronic, Holiday, Singer/Song writer, Jazz, Latino, New Age, Pop, R&B/Urban, Soundtracks, Dance, Hip Hop/Rap, Word, Alternative, Rock, Religious, Vocal, Reggae, Easy Listening |

Media as a starting point to ease the cold start problem.

In this paper, we aim to tackle the above-highlighted downsides to introduce our new music RS.

Our new method can be applied in different real-life situations. For example, as a direct application of this work, we can integrate our solution in an online radio station in order to deal with the challenges raised in (Ignatov, Nikolenko, Abaev, & Poelmans, 2016) such as cold start, boosting of rankings, preference and repertoire dynamics, and absence of explicit feedback. Indeed, the main goal of those stations is to automatically build streamed playlists for the audience. However, with the above problems, online radio stations tend to automatically compose playlists according to a catalog of selection criteria like hits or mood. After the first interaction with these services, the track listening histories of their users are used as input data. With the emergence of mobile devices and the rapid growth of social networking services, contextual information can be easily collected, so our predictive method can be useful in that case.

## 4. Music recommendation challenges

Some of the top-priority challenges, that face RS, are how to deal with data scalability and sparsity, how to solve the cold start problem, and how to behave with special users which require particular reasoning. Further particularities make music recommendation a tricky task and distinguish it from other recommendations like movies and books (Schedl et al., 2018). In this work, we are focused on the following challenges:

- Music recommendation systems are promoted to generate a playlist instead of a single item since music is usually consumed sequentially. So, they need to identify the right ranking of the recommended playlist (Top-N recommendations) and to guarantee its diversity in terms of genres and artists (Lee, 2011).
- Recommending the same previously recommended music track, after a while, may be appreciated by the user of a music RS, unlike a movie or book RS, where repeated recommendations are to be avoided.
- Recommending a playlist of different music genres may be appreciated by users to improve the diversity and the contextualization of recommendations.
- The size of music tracks and albums publicly available, *circa* millions of music pieces, is significantly higher than the size of the movie catalog which counts about thousands of movies. For that reason, the scalability issue could not be overlooked. Based on the fact that humans are naturally expressing their musical preferences by genre granularity, the use of genre items instead of piece items may face the scalability problem without impacting recommendation quality. For example, recommending a Jazz song of *Norah Jones* to a user who is waiting to listen to his/her favorite Jazz singer *Ella Fitzgerald* may not be unappreciated.
- Much information, besides item description and user profiles, needs to be considered while recommending a music item. Indeed, the listening context represents an important factor that may influence music preferences.
- The restriction to the accuracy criteria, when using the contextual information, fails to assess the performance of music RS. The so-called beyond-accuracy measures, like utility, define good means to evaluate contextual recommendations based on questionnaires and user surveys.
- The music RS needs to generate recommendations as soon as the user starts using it. Solutions for the new user cold-start problem require an initial human effort (rating items or answering personal questions) that all users will not appreciate. In such cases, stereotypes, which define users' group interests related to the same context, can help to generate items that users would globally appreciate.

We briefly sketch, in the next section, the preliminary study we conducted to find out the contextual information used to define our contextual model. We consider the selected information important for the recommendation process since they have a relevant contribution to express the variance of users' ratings (Ben Sassi et al., 2017b; Ben Sassi & Ben Yahia, 2021).

## 5. Preliminary study

Before we designed our music CARS, we conducted a preliminary study to find out the information that can influence users' preferences in a music recommendation scenario. The primary objective of our study is to determine the dependency between users' preferences and their context. For example, when they are sad, do they prefer to listen to sad music or to happy songs to get out of their mood? Or do some of them prefer the first category, while others prefer the second style? We have started a user-based study of 109 participants asked to respond to an online questionnaire to express their musical needs (Ben Sassi et al., 2017b). These participants included 59 women (54.1%) and 50 men (45.9%). Their average age is ranged from 17 to 36 (age:17–19: 15; age: 20–29: 86; age: 30–36: 8) with different educational backgrounds (college student: 69; engineer: 21; master student: 6; PhD student: 13). It is worthy of mention that none of them had professional musical training.

### 5.1. User context

An important task in context representation is the identification of

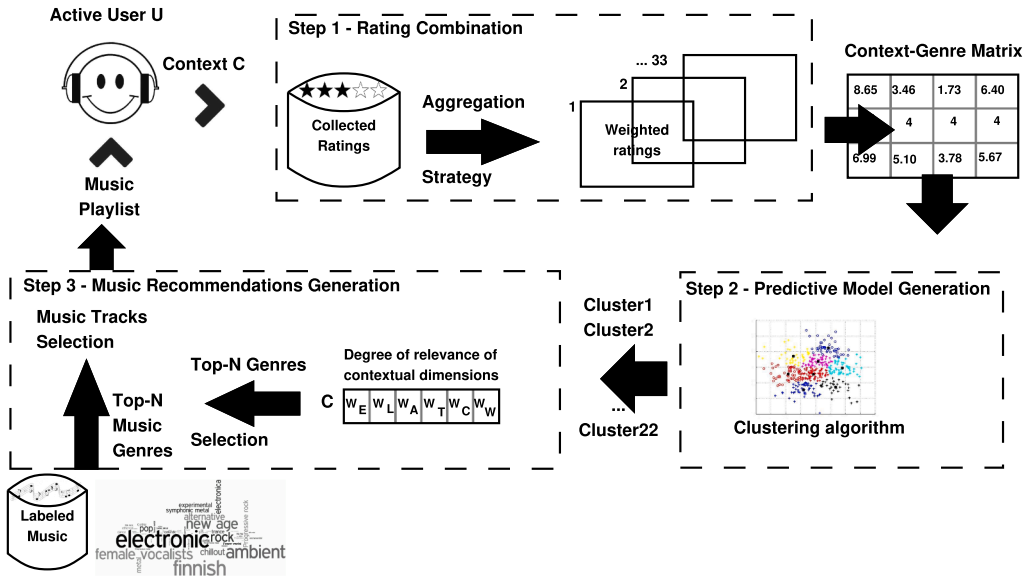



**Fig. 1.** The MORec approach architecture.

the relevant contextual factors, for the specific domain. Indeed, the use of contextual information that does not have a relevant contribution to express the variance of users' ratings, leads to an increase in the recommendation complexity and degrades the prediction quality by adding noise (Baltrunas, Ludwig, Peer, & Ricci, 2012). Hence, we have surveyed previous works on RS literature to extract the most used contextual factors (see Table 1).

We have used the vector model to aggregate our contextual dimensions in a context model. Thus, a context is represented as a set of static factors (denoted as dimensions, e.g., temporal context) with several possible values (denoted as attributes, e.g., part of the day and day of the week) known a priori (e.g., morning and evening). Eq. (1) defines the vector model of the context.

$$\mathscr{C} = (\mathscr{C}_T, \mathscr{C}_L, \mathscr{C}_P, \mathscr{C}_A, \mathscr{C}_E, \mathscr{C}_S) \tag{1}$$

where $\mathscr{C}_T$ (respectively $\mathscr{C}_L, \mathscr{C}_P, \mathscr{C}_A, \mathscr{C}_E$, and $\mathscr{C}_S$) refers to the temporal (respectively location, physical, activity, emotional, and social) context.

### 5.2. Music preferences

Several ways can be employed to express people's musical preferences using various levels of abstraction, such as pieces, albums, artists, or genres of music liked by people. For example, a person can express his interest in special music through a given song, e.g., "Summertime", an artist, e.g., "Ella Fitzgerald", a genre, e.g., "Jazz", a sub-genre, e.g., "Vocal Jazz", or even some music mood attributes, e.g., "Atmospheric" or "Elegant". Consequently, studies have to identify the level of abstraction that will be used to categorize music. The simplest idea is to adopt the level that individuals naturally use to express their musical preferences. Music genres are considered as the optimal level of abstraction to assess people's musical preferences (Schedl & Ferwerda, 2017) since it has the potential to describe any users' tastes. We have noticed that there is no unique categorization of music genres. Thus, to solve this second problem, we have chosen to start with iTunes store[8] music genres and we have validated this set through the focus group technique (Van Eeuwijk & Angehrn, 2017). The analysis of the collected data allowed us to remove some correlated music sub-genres that already exist in other categories, e.g., Comedy is a part of the Vocal genre, and Opera is contained in the Classical genre. In addition, we have changed the Inspirational genre that contains Christian & Gospel to a Religious genre that includes Anachid and religious tracks. Thus, we have classified music tracks in 22 genres (c.f., Table 2).

### 5.3. Study design

The main objective of CARS is about adapting the recommended item to the user's contextual situation. So, they start with the acquisition of item ratings in various possible contextual situations.

We rely on users' perceptions to express the role of context in their decisions. We have opted for the "perceived rating" (or supposed rating) rather than the "actual rating" (or real rating) as it takes less time and because it is very difficult if not impossible to have all users in all possible contextual situations for rating (Ono, Takishima, Motomura, & Asoh, 2009). In addition, we can get various ratings of the same item, given by the same user, in all possible contextual situations, even if (s) he had experienced the item in only one (or few) situation(s). Since online surveys represent an efficient and low-cost way to collect data rapidly, we have developed an online questionnaire. An English version of our questionnaire is available on the following website.[9] Hence, to allow the interviewees to express their degrees of agreement or disagreement versus a given question, we have used the Likert Scale (Likert, 1932). This scale has the advantage that it does not expect a yes/any response, but enables people to precise their degree of opinion, even when they have no opinion at all, in order to collect quantitative and subjective data. This method allows more subtlety in evaluation than in a forced choice. However, respondents can interpret the scale differently from one to another, e.g., one person's three may be equal to another's four. To mitigate this ambiguity, we have annotated scales with labels, i.e., No value = "*I do not know or I do not want to say*", 1 = "*I do not like very much*", 2 = "*I like a little*", 3 = "*I like*", 4 = "*I often like*", and 5 = "*I really like*". So, for each contextual dimension, the participant is asked to evaluate the list of music genres using a Likert preference scale. For example, we can ask the user to pretend that (s) he is using our music RS. We start by asking him/her to assess how likely (s) he will listen to a given music genre, like Jazz, using a 5 rating scale. Next, the same user has to imagine that (s) he is playing sports and we ask him/her again to rate the same musical genre.

[8] http://www.apple.com/itunes.

[9] http://goo.gl/forms/xroRPBH5qs.

**Table 3**
Results of average strategy with equal weights.

| Users/Genres | Genre 1 | Genre 2 | Genre 3 | Genre 4 | Genre 5 |
|---|---|---|---|---|---|
| Robin | 5 | 2 | 1 | 3 | 5 |
| Ted | 1 | 4 | 4 | 4 | 3 |
| Barney | 5 | 2 | 1 | 3 | 4 |
| $\mathscr{AR}$ | 3.7 | 2.7 | 2.0 | 3.0 | 4.0 |

**Table 4**
Results of the average strategy with weights 1.89 for Activity, 1.73 for Time and 1.00 for Weather.

| Context/Genres | Genre 1 | Genre 2 | Genre 3 | Genre 4 | Genre 5 |
|---|---|---|---|---|---|
| $\mathscr{AR}$ | | | | | |
| Time:Morning | 5.0 | 2.7 | 1.0 | 3.7 | 5.0 |
| Weather:Rainy | 1.0 | 4.0 | 4.0 | 4.0 | 3.4 |
| Activity:Relaxed | 3.7 | 2.7 | 2.0 | 3.0 | 4.0 |
| $\mathscr{WAR}$ | | | | | |
| Time:Morning | 8.65 | 3.46 | 1.73 | 6.40 | 8.65 |
| Weather:Rainy | 1.00 | 4.00 | 4.00 | 4.00 | 3.40 |
| Activity:Relaxation | 6.99 | 5.10 | 3.78 | 5.67 | 7.56 |

### 5.4. Contextual factors relevance

We have analyzed the participants' responses to unveil their experiences with music and concretize the effect of contextual situations on the type of listened music. From our preliminary study, we found users treat contextual information in different ways, giving more importance to some contextual information than the other ones. The order of importance between contextual information is defined as $Emotion \succ Location \succ Activity \succ Time \succ Companion \succ Weather$.

Based on the Multi Linear Regression technique (Draper & Smith, 1998), we learned the importance of context dimensions for music decision making. We derived the following weights: $W_{Emotion} = 0.323$, $W_{Location} = 0.228$, $W_{Activity} = 0.134$, $W_{Time} = 0.123$, $W_{Companion} = 0.121$, and $W_{Weather} = 0.071$ from the data collected via the online questionnaire. Based on these weights, we can derive that the context dimension Emotion (respectively Location, Activity, Time, and Companion) is 4.55 (respectively $3.21, 1.89, 1.73$, and 1.70) times as important as Weather.

## 6. Context-aware recommendation algorithm

In this work, we focused on integrating MCRS and CARS to solve our music recommendation problem. As mentioned in (Adomavicius et al., 2011), there are different ways to build multi-criteria recommendation algorithms. The aggregation-based method is one of the most effective approaches. In this section, we introduce our approach based on the aggregation method. More precisely, the problem that has to be addressed includes how to proceed with the high dimensionality of our problem? How to deal with new users or new items? How to consider the different user rates of every music genre in each contextual situation? and how to select the most suitable music genre to be recommended, which satisfies multiple contextual situations?.

In the following, we describe our new recommendation approach, named *MORec* (Music Online Recommendation) allowing the generation of clusters of similar contexts to compute music genres recommendation (c.f., Fig. 1). Thus, our approach operates through three steps: *(i) Rating combination*: We apply a rating combination technique which aims to aggregate information about individual likes and dislikes in order to compute the overall rating; *(ii) Clustering-based predictive model generation*: We apply the K-means algorithm to generate clusters of similar contexts defining the relation between contextual dimensions and music genres; and *(iii) Context-Aware Music Recommendations Generation*: We compute the matching between the user's current context and the generated clusters to rank the list of music genres that will be recommended to him/her. We detail these steps in the following.

### 6.1. Rating combination

Multiple aggregation strategies have been applied in the past to tackle recommendation problems (Masthoff, 2011). In a music recommendation scenario, the user may prefer a music genre that motivates him/her to accomplish tasks and activities. (S) he may also prefer another music genre that makes him/her feel better. In other terms, individual interests are multiple, heterogeneous, changing, and maybe influenced by multiple factors. Here, the RS possesses music genres rated differently in each criterion or contextual information. Thence, the RS has to select the most suitable music genre to be recommended, which satisfies multiple criteria.

#### 6.1.1. Average aggregation

We propose to combine the individual ratings by the average strategy since it has given encouraging results in (Masthoff, 2011). The average aggregation method averages individual ratings as expressed in Eq. (2).

$$\mathscr{AR} = \frac{1}{n}\sum_{i} R_i \tag{2}$$

where $\mathscr{AR}$ denotes the average rating, $R_i$ stands for the individual ratings, and $n$ is the number of individuals.

Suppose we have three users: Robin, Ted, and Barney. Table 3 shows an example of ratings of five music genres, on a scale of 1 to 5, of these three users. It also gives the result of the average strategy with equal weights applied to the users' ratings. The computed average ratings will lead to rank the five genres to be recommended as {Genre 5; Genre 1; Genre 4; Genre 2; Genre 3}.

#### 6.1.2. Weight-based criteria

When aggregating ratings, the totality of aggregation methods has treated all people equally without discrimination. In our work, we are interested in contextual information aggregation. When adapting to multiple criteria, which we define as contextual information, there is no reason for considering all criteria with the same importance. Indeed, based on our preliminary study detailed above, we have found that contextual information influence differently users' music preferences. We apply the weight-based criteria method. The weight of the contextual information, derived from our preliminary study, is multiplied by its average ratings to yield "weighted" average ratings. We propose to use the weights as multipliers, leading to Eq. (3).

$$\mathscr{WAR} = \frac{1}{n}\sum_{i} W_i R_i \tag{3}$$

where $\mathscr{WAR}$ stands for the weighted average rating, $R_i$ for the individual ratings, $W_i$ the weight criterion, and $n$ is the number of users. Table 4 shows the result of the average strategy, related to three criteria using the weights, detailed in Sub-Section 5.4.

At the end of the rating combination step, we get a $M \times N$ Context-Genre Matrix, where $N$ is the number of musical genres and $M$ is the number of contextual dimensions. The Context-Genre Matrix defines the preferred music genres in each contextual dimension. For example, based on the example detailed in Table 4, the most preferred genres in the contextual dimension value *Time:Morning* are Genre 1 and Genre 5 with a rating equal to 8.65, followed by Genre 4 and Genre 2 with the respective ratings 6.40 and 3.46. Nevertheless, Genre 3 having 1.73 as a rating value is seen as the less preferred genre.

### 6.2. Clustering-based predictive model

In this step, the above-formed matrix, describing the relation between contextual dimensions and the aggregated ratings, is used for

similar contextual dimensions clustering.

#### 6.2.1. The K-means algorithm

The K-means is a simple, quick, and relatively efficient unsupervised clustering algorithm. However, it has several limitations. First, it assumes a certain beforehand knowledge about data to choose the optimal number of centroids that will affect the generated clusters. Second, sometimes, this algorithm generates empty clusters. In our case, we are not concerned with the above shortcomings, since we have genuine knowledge about the data presented by the Context-Genre Matrix formed in the first step of our approach. Alternatively stated, we can select the number of centroids $k$, which represents the number of music genres.

#### 6.2.2. Similar contextual dimensions clustering

Whenever a new user enters the system for the first time, the state-of-the-art RSs, relying on clustering techniques (items or users clustering), cannot generate recommendations, since they do not have previous information about him/her. To tackle the cold start problem, we have proposed a clustering-based approach that does not process individual ratings, but rather average ratings. This step generates clusters by applying the K-means clustering algorithm, which takes as input the weighted average ratings modeled as a Context-Genre Matrix. We clustered contextual dimensions based on the weighted-average ratings aggregated in the first step. Each contextual dimension is represented, in the Context-Genre Matrix, as a row that contains the weighted-average ratings given to the music genres in this particular contextual dimension value. The output of this step is $k$ clusters of similar contextual dimensions related to each music genre. Thence, similar contextual dimensions that have similar average ratings are gathered in the same cluster. For contextual dimensions clustering, we adopt the typical K-means algorithm of (Hartigan & Wong, 1979), which represents an alternative heuristic of the classical algorithm aiming to optimize the K-means cost function (Telgarsky & Vattani, 2010).

The clustering method based on the K-means algorithm operates as detailed in Algorithm 1. It is worth mentioning that the convergence criterion (line 6) is met whenever the centroids stop changing. More precisely, the algorithm repeats the 2 steps (c.f., lines 4–5) until no contextual dimension changes its cluster.

**Algorithm 1**. Clustering-based Predictive Model

```
   Data: CGM: Context-Genre Matrix;
         K: Number of genre clusters.
   Result: clust: Set of genre clusters.
1  begin
2      Select K initial centroids, which represent the
         number of music genres;
3      repeat
4          Create K genre clusters by assigning each
             contextual dimension to the cluster that
             has the closest centroid;
5          Update the positions of the K centroids by
             computing the mean of all clusters;
6      until Convergence criterion is met;
7  end
```

### 6.3. Context-aware music recommendations generation

Given a user $u$ along with a context defined as $\mathscr{C} = (\mathscr{C}_T, \mathscr{C}_L, \mathscr{C}_P, \mathscr{C}_A, \mathscr{C}_E, \mathscr{C}_S)$. The goal of this step is to select the list of Top-N genres that matches with the user's context $\mathscr{C}$. We first locate the clusters, which exhibit the current context $\mathscr{C}$ of the user $u$ from the set of clusters previously generated. So, not all clusters are relevant as some of them describe different contextual dimension values than those defining $\mathscr{C}$. As far as the selected clusters, denoted as $clust_u$, must contain one dimension of the current context of $u$, then their number cannot exceed 6 clusters, which define the size of the user's context. Besides, because of the different degrees of relevance of contextual dimensions, not all clusters of $clust_u$ are equally relevant. Therefore, we have to assess the relevance of the clusters of $clust_u$ regarding the context $\mathscr{C}$. In doing so, we compute the matching between the user's current context $\mathscr{C}$ and the generated clusters $clust_u$ based on the relevance order of contextual dimensions, inferred in Sub-Section 5.4 and expressed as $\mathscr{C}_E \succ \mathscr{C}_L \succ \mathscr{C}_A \succ \mathscr{C}_T \succ \mathscr{C}_S \succ \mathscr{C}_P$. In other words, we select all genre clusters that contain any of the user's context dimensions. The selected clusters are then ranked, using the importance of the contextual dimensions, to build the Top-N music genre recommendations. For instance, the genre associated with the cluster that contains the user's emotional context $\mathscr{C}_E$ is ranked at the top of the Top-N list, followed by the one which contains his/her location, and so on. Finally, the ordered list of music genres is used to recommend a playlist of music tracks, related to these genres. It remains to define the optimal number of music tracks into a playlist. The playlist length depends on what kind of playlist is in progress, and the device used to listen to it. The minimum number of playlist tracks in Deezer is about 30. Genius puts together 25 songs to generate its playlists. In the digital music service Spotify, the user can access a Discover playlist, refreshed weekly, which contains 30 songs (about 2 h) of unheard or similar music (s) he listened to. The top 10 playlists on Spotify have an average of 58 tracks. So, we choose to recommend playlists with a maximum of 48 tracks, i.e., 8 songs by genre. These 48 music tracks belong to the Top-N music genres previously generated that aims to guarantee the diversity of our recommendations in terms of genres. Based on the assumption that users may appreciate receiving the same music piece many times, we have randomly extracted music items from the set of songs. As a result, we may extract again the same music item after several executions. In addition, in terms of a variety of artists, our selected recommendations will help users to discover new artists.

The pseudo-code of the MORec approach is sketched by Algorithm 2. The recommendation algorithm ushers by computing the Context-Genre Matrix using the *WeightedAverageAggregation* function invoked in line 3. In line 4, we invoke the *ClusteringbasedPredictiveModel* function, described in Algorithm 1. The latter generates the predictive model based on the K-means algorithm. In line 13, we generate a list of Top-N genres, denoted $G$, that matches the user's context $C_u$. Then, the obtained Top-N list, in line 14, is used to extract a list of music tracks. In our work, we set the parameter *size*, defining the number of tracks by music genre, equal to 8.

**Algorithm 2**. MORec: Online Music Recommendation

**Data:** $R$: Users ratings;
$C_{ord}$: Contextual dimension relevance order;
$C_{rel}$: Contextual dimension relevance degree;
$clust$: Set of genre clusters;
$C_u$: Context of the target user;
$MDB$: Music tracks database.
**Result:** $G$: Top-N recommended genres;
$T$: Set of the recommended music tracks.

**begin**
  $AR$= AverageAggregation($R$);
  $CGM$=WeightedAverageAggregation($AR$,$C_{rel}$);
  $clust$= ClusteringbasedPredictiveModel($CGM$);
  $clust_u = \emptyset$;
  **foreach** $dim \in C_u$ **do**
    **foreach** $cl \in clust$ **do**
      **if** $dim \subset cl$ **then**
        $clust_u = clust_u \cup cl$;
      **end**
    **end**
  **end**
  $G$= TopNGenresRanking($clust_u$, $C_{ord}$);
  $T$= MusicTracksSelection($G$, $MDB$, $size$);
**end**

### 6.4. Illustrative example

Let us consider a user working in a computer society. On a sunny day of September, before entering the meeting scheduled at 10 : 00 AM with his supervisors, this user was listening, via an online mobile App, while working on a playlist of Pop music. We can define the context of the target user as $\mathscr{C} = (\mathscr{C}_T, \mathscr{C}_L, \mathscr{C}_P, \mathscr{C}_A, \mathscr{C}_E, \mathscr{C}_S) = (morning-wokday, work, sunny, reflection, joy, with-colleagues)$. After his meeting, the user felt so angry and shared a tweet $\mathscr{T}$ saying *"Going to freak out really soon! -_- I honestly can't wait to get out of here next year #annoyed #pissedoff"*. Thus, after the switching of the user's emotional state from "joy" to "anger", his context will be updated and is the equal to: $\mathscr{C} = (morning-wokday, work, sunny, reflection, anger, with-colleagues)$.

Here, based on our predictive model, developed in Sub-Section 6.2, the selected list of genres that match the user's context is as follows: {Rock; Blues; Soundtracks; Classic; Electronic; Pop}. The generated Top-N genres list gathers various music genres, where each genre is associated with one of the contextual dimensions which define the user's context $\mathscr{C}$, and are ranked by contextual dimensions importance. Plainly speaking, the first item of the list "Rock" is associated with the target user's emotional context "anger", the second one "Blues" is related to his location "work" and so on. The next step of our MORec approach is the extraction of random tracks associated with the generated Top-N genres list (8 tracks by genre). We describe the recommended playlist in Table 5. As shown, for each genre, the recommendable songs to this target user are diverse in terms of artists and sub-genres. For example, for the "Rock" music genre, the recommended tracks belong to various artists and four different sub-genres of "Rock": "Rock Pop", "Hard Rock", "Punk", and "Country Rock", to meet the maximum of users' expectations and allow them to discover new artists they do not know. We can also note the diversity of the Top-N selected music genres. Indeed, the generated playlist contains high arousal music like "Rock", "Electronic" and "Pop", which will be appreciated by the target user if he prefers listening to anger and rhythmic music when he is irritated. Besides, it contains "Blues" and "Classic" music to help him, need be, to calm down and return to work.

## 7. Experimental evaluation

The purpose of our experiments is to evaluate the suitability of the genre of music tracks recommended by our approach and their usability. We start by evaluating our prediction model, which is based on the partitional clustering technique. Then, we evaluate our recommendation system based on a user-centered method, from two main perspectives, i.e., from quality and usability respective angles. Finally, we compare our recommendation approach, in terms of precision and recall, versus both traditional and context-aware models.

### 7.1. Datasets description

First, the RS was trained with the rating dataset described in Section 7.1.1. Next, the set of music tracks, related to each music genre, constituting potential recommendations, was extracted from the MusiClef dataset well known in the music field and described in Section 7.1.2. Finally, the test participants (11 users studying at the Faculty of Sciences of Tunisia) have answered a set of questions related to our recommendation approach results (see Section 7.1.3). The three used datasets are presented in the following.

#### 7.1.1. COMUDA: A context-aware music dataset

In order to be able to generate our predictive model for context-aware music recommendations, we need a contextual dataset that associates users' ratings to the different contextual situations. However, available datasets for music filtering tasks, like Yahoo! Music or Last.fm only contain temporal information extracted from timestamps or general information about users such as gender, age, country, sign-up date, which cannot be translated to contextual information. Since we were concerned with the most used contextual factors in RS literature, we have decided to collect our own data in order to create a contextual music dataset. As we have detailed in Section 5, we have collected item ratings in various possible contextual situations. More details about the collected dataset are given in Appendix B.1.

**Table 5**
The recommended music tracks ranked by the Top-N genres.

| |
|---|
| *Top 1: Rock* |
| JUMP IN THE RIVER - SINEAD O'CONNOR |
| PERSONALITY CRISIS - NEW YORK DOLLS |
| FIDO - BYRDS |
| DREAM ON - AEROSMITH |
| HAVING A BLAST - GREEN DAY |
| HARVEST FOR THE WORLD - ISLEY BROTHERS |
| EVIL WOMAN - BLACK SABBATH |
| I CORINTHIANS 15 : 55 - JOHNNY CASH |
| *Top 2: Blues* |
| I GOT THE BLUES - SOLOMON BURKE |
| DO WHAT YOU DO - TINA TURNER |
| WHY DO YOU DO ME - JAMES BROWN |
| COME AND GET THESE MEMORIES - VANDELLAS |
| YOUR LOVE HAS BROUGHT ME A MIGHTY LONG WAY - WILSON PICKETT |
| IT'S ALRIGHT - RAY CHARLES |
| COUNTRY WOMAN - BEE GEES |
| GOD MADE YOU FOR ME - AARON NEVILLE |
| *Top 3: Soundtracks* |
| THE LORD OF THE RINGS: THE TWO TOWERS - HOWARD SHORE |
| BATMAN BEGINS - HANS ZIMMER |
| STAR WARS - JOHN WILLIAMS |
| THE LEGEND OF ZELDA: OCARINA OF TIME - KOJI KONDO |
| THE LORD OF THE RINGS: THE FELLOWSHIP OF THE RING - HOWARD SHORE |
| THE LORD OF THE RINGS: THE RETURN OF THE KING - HOWARD SHORE |
| THE LION KING - SALLY DWORSKY |
| CONAN THE BARBARIAN - BASIL POLEDOURIS |
| *Top 4: Classic* |
| YOUR BABY DOESN'T LOVE YOU ANYMORE - CARPENTERS |
| HOW TO FALL IN LOVE - BEE GEES |
| WE'LL TAKE THE NIGHT - ROY ORBISON |
| SECRET GARDEN - BRUCE SPRINGSTEEN |
| SAVOIR FAIRE - CHIC |
| BEN ESCAPES - HOLLYWOOD STUDIO SYMPHONY |
| JESSE - ROBERTA FLACK |
| LONG AGO AND FAR AWAY - JAMES TAYLOR |
| *Top 5: Electronic* |
| MAKE LOVE - DAFT PUNK |
| DIG FOR FIRE - PIXIES |
| SAY IT LOUD - AFRIKA BAMBAATAA |
| AMAZON (RIVER OF DREAMS) - BAND |
| CAN'T GIVE YOU UP - AFRIKA BAMBAATAA |
| FAMILY AFFAIR - SLY & THE FAMILY STONE |
| DRAGON QUEEN - YEAH YEAH YEAHS |
| CRESCENDOLLS - DAFT PUNK |
| *Top 6: Pop* |
| THIS FIRE - FRANZ FERDINAND |
| JUST GET UP AND DANCE - AFRIKA BAMBAATAA |
| FOOTSTEPS IN THE DARK - ISLEY BROTHERS |
| THIS TIME - VERVE |
| HARDER BETTER FASTER STRONGER - DAFT PUNK |
| NEVER TOO LATE - KYLE MINOGUE |
| LA LA LA HE HE HEE - PRINCE |
| LET DOWN - RADIOHEAD |

#### 7.1.2. MusiClef Dataset: MIR:MusiClef:2012: MMSys: version1.0

The MusiClef dataset[10] contains multimodal data of professionally annotated music related to 1355 popular music songs by 218 artists (Schedl, Liem, Peeters, & Orio, 2013). We opted for this dataset, instead of other publicly available annotated music datasets like lastfm-2 k or Million Song datasets, since it was used as a standard dataset for multimodal music retrieval task in MusiClef 2011 and MusiClef 2012 evaluation campaigns and was been proven relevant to a real-life use case. More details about this dataset are given in Appendix B.2.

#### 7.1.3. Diary test dataset

11 users participated in our evaluation task. Indeed, the author in (Macefield, 2009) have found that there is no one test group size which can fit all the problem. But, for studies related to problem discovery a group size of 5–10 participants is a sensible baseline range. In this work, we ran our tests with 11 participants heterogeneously distributed by age, gender and educational background. More details about this dataset are provided by Appendix B.3.

[10] http://www.cp.jku.at/datasets/musiclef.

**Table 6**
User-based evaluation of the recommendation performance.

| Metrics | NDPM | User Coverage | Item Coverage | Diversity |
|---|---|---|---|---|
| MORec | 0.408 | 1.000 | 0.236 | 0.818 |

**Table 7**
Recommendation performance comparison.

| Approaches | P@6 | R@6 | F@6 |
|---|---|---|---|
| *Traditional Recommendation Algorithms* | | | |
| *Average Baselines* | | | |
| GlobalAverage | 0.151377 | 0.427471 | 0.223579 |
| UserItemAverage | 0.162476 | 0.472328 | 0.241781 |
| *Top-N Ranking Baselines* | | | |
| BPR | 0.094388 | 0.349884 | 0.148669 |
| LRMF | 0.154512 | 0.454951 | 0.230679 |
| *Collaborative Filtering* | | | |
| SVD++ | 0.161620 | 0.485846 | 0.242553 |
| BPMF | 0.158205 | 0.468772 | 0.236570 |
| *Context-Aware Recommendation Algorithms* | | | |
| *Transformation Algorithms* | | | |
| SPF | 0.087415 | 0.415057 | 0.144414 |
| UISplitting | 0.160592 | 0.484999 | 0.241288 |
| *Adaptation Algorithms* | | | |
| FM | 0.130653 | 0.363101 | 0.192161 |
| CAMF_C | 0.158396 | 0.466218 | 0.236456 |
| CAMF_CI | 0.160119 | 0.476614 | 0.239707 |
| CAMF_CU | 0.153393 | 0.432240 | 0.226430 |
| CAMF_CUCI | 0.156603 | 0.469557 | 0.234872 |
| CSLIM_C | 0.153202 | 0.439678 | 0.227228 |
| CSLIM_CI | 0.150460 | 0.433574 | 0.223396 |
| CSLIM_CU | 0.004365 | 0.026190 | 0.007482 |
| Our Algorithm | | | |
| MORec | 0.407000 | 0.573000 | 0.475940 |

### 7.2. Clustering-based model evaluation

Finding the best data representation is one of the typical unsupervised scenario challenging problems. As a result, defining clustering validation measures presents an important topic for clustering approaches. Validation measures can be categorized into three types: internal, external, and relative (Rendón, Abundez, Arizmendi, & Quiroz, 2011). The use of external criteria implies the presence of the ground truth defined by a pre-specified structure. In our context, this information is not present in the dataset. So, we opted for the evaluation based on internal and relative criteria. Internal indices evaluate the clustering results with respect to information only obtained from the data. These criteria assign best values to clustering approaches that generate clusters with high similarity between inter-cluster objects and low similarity between intra-cluster objects. However, relative criteria are based on multiple clustering results comparison. We describe, in Appendix C, different experiments we conducted for our predictive model evaluation. We start by executing the K-means algorithm with different values of $K = 3, 6, 10, 15, 22, 23, 24, 25$ to compute several internal indices. We prove that, for the majority of indices, the best results are obtained with

$K = 22$. Then, we compare the K-means algorithm with the Partitioning Around Medoids (PAM) algorithm (Kaufman & Rousseeuw, 1990), for $K = 3, 6, 10, 15, 22$ clusters, in terms of connectivity and stability.

### 7.3. Context-aware recommendation evaluation

In the following section, we detail the experiments we conducted to evaluate MORec's performances based on a user-centered method and we compare them versus those of the state of the art baselines.

#### 7.3.1. User-based evaluation

User-based evaluation has been commonly used to enrich the evaluation of RS. We measure the user satisfaction when the knowledge of contextual variables is exploited and without considering any contextual information.

The goal of our recommendation approach is to provide a target user a vertical list of items presented with a certain order, known as *ranking items task*. Indeed, MORec aims to recommend a set of Top-N music genres that match the target user's contextual situation. For each genre, a list of music tracks are randomly generated to be recommended to him/her. So, we are not simply interested in recommending the most interesting items like the majority of RS, but rather in ordering Top-N items based on the context of the target user.

The results of the MORec approach based on the ranking and usefulness evaluation are depicted in Table 6.

***Top-N Recommendation Ranking Evaluation*** represents an interesting metric to assess the accuracy of the ranking items task. We opted for the Normalized Distance-based Performance Measure (NDPM) (Shani & Gunawardana, 2011) commonly used to evaluate information retrieval systems by comparing the order of ranking of two documents. The NDPM metric is defined by Eq. (4). Its highest score equal to 0 underscore that the system can correctly predict the totality of preference relations expressed through the ranking references. However, it yields 1 whenever the system contradicts all reference preference relations.

$$NDPM = \frac{2C^- + C^+}{2C^*} \tag{4}$$

where $C^-$ is the number of contradictory preference relations that occur when the system ranks that item $a$ will be preferable to item $b$, even, the user's ranking is the opposite; $C^+$ is the number of compatible preference relations that happen when item $a$ is ranked better than item $b$ in the system ranking, whereas, the user considers them equal; and $C^*$ is the total number of relations of item pairs, having the same order, between the system and the user ranking.

Based on the test dataset, which contains users' explicit ratings, we generated the reference ranking with the correct order of items. We selected the 10 most frequent contextual situations and in each one we have compared the Top-N genres recommended by MORec to the participants' actual rankings. Next, we have applied the NDPM metric for each user: $NDPM(i), i = 1, \ldots, 11$. Finally, we have computed NDPM, the average value of NDPM(i) related to all users of the test dataset.

The average NDPM metric score, for our test dataset, is about 0.408. This value can be explained by the particular preferences of some users. For example, for the contextual situation $\mathscr{C} = (morning - workday, work, sunny, reflection, anger, with - colleagues)$, the user $U7$ has asserted that she only prefers to listen to rock music with a rating equal to 5. So, even if MORec has recommended rock music in the first position, the rest of the recommendations, related to the 5 other genres, are unneeded. In this case, the recommendation of the other 5 genres has decreased the quality of the recommended list, even with the presence of the most preferred music genre at the top of the list.

***Usefulness Evaluation: Coverage and Diversity*** Besides recommending items with good quality in terms of ranking, we intend to satisfy usefulness metrics that assess the suitability of recommendations to users.

**Genre Coverage** is the most evident property. It refers to the portion (or the percentage) of resources that a referral system can recommend (Shani & Gunawardana, 2011). Since, most users prefer recommendations related to a variety of genres (Lee, 2011), it is important to cover as many music genres as possible. Indeed, recommendation algorithms need to make sure to recommend items with high quality and, at the same time, cover a wide span of genres. In addition, a new constraint, called "recommendation list size-awareness" (Vargas, Baltrunas, Karatzoglou, & Castells, 2014), is added with the context of the mobile environment. Indeed, the limited screen space of mobile devices needs to be considered. Thus, we can not talk about the coverage of a list of recommendations without taking into account the size of the list proposed to the target user. MORec selects a set of 6 genres, which match the user's context, to recommend a playlist of 48 music tracks. We used the 10 contextual situations, previously selected in the quality evaluation, and in each one, we computed the coverage measure value. To do so, for each situation, we divided the number of the distinct selected genres by the total number of genres in our dataset. The average coverage of MORec is about 0.236.

**User Coverage** defines the portion of users for which the system can recommend items (Shani & Gunawardana, 2011). Algorithms capable of generating recommendations for the majority of users and especially for particular ones (with few or no ratings) are therefore particularly appreciated. In our case, MORec provides recommendations based only on the user's current context not on other personal information about him/her that can restrict the generated recommendations to his/her similar users. Indeed, MORec is able to provide recommendations to (almost) all users regardless of whether they have previously provided ratings or not. Indeed, it uses stereotypes that define users overall interests related to the same context. Among the 11 users of the test dataset, MORec reaches a coverage equal to 100%.

**Diversity** is the opposite of similarity (Shani & Gunawardana, 2011). It also measures the ability of a system to avoid the so-called filter bubble issue (Haim, Andreas, & Brosius, 2018). Sometimes, users need to receive different items in order to explore the totality of items, with minimal impact on accuracy. Like a user of a trip RS who needs to get various POIs for his/her vacation. Genre information was considered as a powerful mean to measure and enhance the diversity of recommendations (Vargas et al., 2014). So, in our case, we are interested in diversity in terms of music genres. MORec provides a list of recommendations which belongs to a set of genres (Top-N music genres). In addition, since we select music songs only based on genres, MORec guarantees diversity in terms of a variety of artists. Indeed, the selection of random songs related to some music genres, which correspond to their contexts, can help users to discover new artists.

However, we need to translate this intuitive diversity to an expressive value based on a mathematical equation. The recommendation diversity can be computed through two metrics (Adomavicius & Kwon, 2012): *(i) individual diversity* is the average of dissimilarity between all pairs of recommended items in order to avoid recommending too similar items for the same user; and *(ii) aggregate diversity* computes the recommendation diversity across all users. In our case, we intend to compute the diversity of our recommendation approach based on the Top-N genres used to generate music tracks to be recommended to the target user. So, we used the aggregate diversity measure proposed by (Adomavicius & Kwon, 2012), which computes the total number of distinct items recommended to all users (see Eq. (5)). We used the 10 contextual situations, previously selected in the quality evaluation, and for each one, we computed the diversity measure value.

$$diversity - in - top - N = |\bigcup_{u \in U} T_N(u)| \tag{5}$$

where $T_N(u)$ is the set of Top-N genres recommended to the user $u$ with $T_N(u) = \{G_1, G_2, \ldots, G_6\}$, and $u$ is a user from our test set. We tested the diversity metric on the 11 users of the test dataset and we obtained an

average diversity score equal to 0.818.

***Cold Start Evaluation.*** The purpose of this experiment is to discuss whether the use of stereotypes, which define users' overall interests related to the same context, can face the cold start problem. Indeed, our prediction model, c.f., Sub-Section 6.2, is based on a Context-Genre Matrix that defines the music genres preferred in each contextual situation. The proposed prediction model is not a personalized model that depends on the target user past ratings. It uses weighted average ratings aiming to provide the same recommendations for all the users sharing the same context. As a result, our system does not suffer from the cold start pitfall.

In order to validate our hypothesis and evaluate the effectiveness of adding contextual information for delivering recommendations to new users, we have compared the results of our approach based on a context-aware predictive model, which uses in-context ratings versus the results of our approach based on a predictive model that uses no-context ratings. As underscored in Section 6, our approach does not consider how a particular user rates a music genre to generate the list of items to be recommended. Plainly speaking, all users having the same contextual situation will receive the same recommendations (not the same music items but the same Top-N music genres). The two variants of our approach that we have compared can be distinguished based on the nature of ratings used to generate the predictive model. The first variant, called MORec relies on the in-context ratings to exploit the knowledge of the contextual information to adapt the music genre recommendations to the contextual situation of the target user. The second variant is based on the overall evaluation of the target population expressed by the average of their no-context ratings to generate an overall ranking of the Top-N interesting music genres. Next, we have compared the results obtained through these two variants with the ratings extracted from the test dataset.

Our results show that recommendation lists generated by MORec are more suitable to meet user's sought after rankings than the recommendations resulted from the popularity-based approach. Indeed, 73% of the test set users have expressed rankings that match the MORec generated list of genres. However, 27% of users have provided ratings similar to the list of recommendations obtained with the popularity-based approach. So, they may prefer non-contextual recommendations more than contextual ones. Probably, the obtained results can be explained by some users who only listen to one music genre in all situations. A larger evaluation may be useful to understand the reason behind these choices.

#### 7.3.2. Assessment of recommendation quality

In the following, we detail our experiments for evaluating the results of our approach against the state of the art recommendation approaches including both traditional recommendation and context-aware recommendation algorithms.

***Baseline Approaches.*** To validate the effectiveness of MORec, we compared it with the several approaches from the CARSKit-v0.3.5[11] Java-Based Context-aware Recommendation Library updated in 2019 (Zheng, Mobasher, & Burke, 2015). The selected toolkit approaches can be classified into two categories based on the information they use.

**Traditional recommendation algorithms:**

- **GlobalAverage**: this algorithm belongs to average RS. It returns the constant rating, i.e., the global average rating from a data set.
- **UserItemAverage**: is an average RS, which returns and average rating by a specific user on a specific item.
- **BPR**: the *Bayesian Personalized Ranking* is a baseline for Top-N ranking recommender based on stochastic gradient descent with bootstrap sampling of training triples.
- **LRMF**: the *List Rank Matrix Factorization* is a Top-N ranking RS which combines a list-wise learning-to-rank algorithm with matrix factorization.
- **SVD++**: is a derivative model of SVD (Singular Value Decomposition), which is one of the common matrix factorization methods used in collaborative filtering. SVD++ achieves better predictive accuracy than SVD owing to the incorporation of implicit feedback information.
- **BPMF**: the *Bayesian Probabilistic Matrix Factorization* algorithm belongs to collaborative filtering systems. It applies Markov Chain Monte Carlo methods to perform approximate inference.

**Context-Aware recommendation algorithms:**

- **SPF**: the *Semantic Pre-Filtering* algorithm belongs to context-aware recommendations based on transformation algorithms. It tries to convert the multidimensional recommendation problem into a traditional 2-dimensional one, so that the traditional recommendation algorithms can still be used.
- **UISplitting**: is a transformation context-aware recommendation algorithm that uses the t-test on the mean rating within two different contextual conditions for the combined user and item splitting task. It splits both users and items in the dataset to boost context-aware recommendations.
- **FM**: the *Factorization machines* is a context-aware recommendation method based on the adaptation finer-grained algorithm that exploits pairwise relationships during its learning process.
- **CAMF**: the *Context-Aware Matrix Factorization* algorithm is a context-aware recommendation method based on the adaptation algorithms, which directly incorporate contexts into the prediction function. We tested four variants of dependant derivation-based CAMF, aiming to model contextual rating deviations: *(i)* CAMF_CI: associates a rating deviation for each pair of item and context condition; *(ii)* CAMF_CU: uses a rating deviation for each pair of user and context condition; *(iii)* CAMF_C: uses a rating deviation for each context condition; and *(iv)* finally CAMF_CUCI: uses a rating deviation for each pair of user and context condition, as well as each pair of item and context condition.
- **CSLIM**: the *Contextual Sparse Linear Method* also belongs to the context-aware recommendation method based on adaptation algorithms. We selected three different implementations of CSLIM algorithm built upon the SLIM_I algorithm. *(i)* CSLIM_C: associates a rating deviation for each context condition; *(ii)* CSLIM_CI: uses a rating deviation for each pair of item and context condition; and *(iii)* CSLIM_CU: uses a rating deviation for each pair of user and context condition.

**Evaluation Metrics.** In our experiments, we examine the precision, recall and F-measure of the evaluated recommendation approaches. As we know, P@k (short for Precision@k), R@k (short for Recall@k), and F@k (short for F-measure@k) are three evaluation metrics for ranking learning.

**Precision@k:** defines the proportion of recommended items in the top-k set that are relevant (c.f., Eq. 6).

$$P@k = \frac{\#relevant\_recommended\_genres@k}{\#recommended\_genres@k} \tag{6}$$

**Recall@k:** measures the proportion of relevant items found in the top-k recommendations (c.f., Eq. 7).

$$R@k = \frac{\#relevant\_recommended\_genres@k}{\#all\_relevant\_genres@k} \tag{7}$$

**F-measure@k:** combines the precision and the recall. Indeed, while maximizing the precision, defined by the ability of the system to estimate positive predictions over the Top-N recommendations, we may hurt the recall, determined by the power of generating positive

[11] https://github.com/irecsys/CARSKit.

predictions with respect to the list of relevant items to the target user (c. f., Eq. 8).

$$F@k = 2.\frac{P@k.R@k}{P@k + R@k} \tag{8}$$

Note that we compute, in this paper, the results of the three evaluation metrics with the setting of $k = 6$, since we are interested in Top-6 recommendations.

***Dataset Preparation.*** As mentioned in (Zheng et al., 2015), two formats can be used to store the contextual rating data, i.e., loose format and compact format. In our case, we opted for the former, assuming that there is only one rating for each pair of user and item in the associated contexts. Indeed, the collected COMUDA and Diary Test datasets contain a single user's rating for each contextual dimension value. Next, we used the CARSKit, and specifically, the *TransformationFromLooseToBinary* method, to convert both datasets to a binary format.

***Experiment Results.*** The performance comparison of our approach and the baseline models is shown in Table 7. First, our proposed model achieves the best performance regarding the three evaluation metrics. More precisely, our model outperforms the other ones, especially in terms of P@6, which reaches 0.407000 with MORec and does not exceed 0.162476 with *UserItemAverage* baseline that yields the best result in terms of precision. One possible reason, cluing the outperformance of our approach compared to the baseline models, is that we are fetching for genres that fit an active user's context only into clusters his/her context belongs to. Thus, the selection of genres to be recommended for a particular user is limited to some clusters which may enhance the Top-N precision by reducing the number of recommended genres, used to compute the P@k metric. Second, in terms of recall, the majority of approaches have yielded approximately equal results, with an improvement of 18.14% for our approach compared to the *UISplitting* algorithm. We can also underscore that the F-measure results are significantly improved by fusing context awareness and MCDM. Indeed, the MORec approach has improved the F-measure results with 97.25% compared to the *UISplitting* algorithm.

## 8. Conclusion and future work

Music recommendation is a research topic that gained increasing interest after the development of online music platforms. However, to provide valuable recommendations it is necessary to deal with the common challenges of recommender systems and those related to the musical application domain, like the difficulty of extracting, modeling, analyzing content information, and dealing with the extensive music collections. In this work, we dealt with some of the music recommender systems' shortcomings, especially the cold start, the diversity, and the scalability problems without needing user attributes, music features, and explicit ratings from users.

The approach proposed, in this work, heavily relies on the beforehand collected data expressing the relationship between users, their context, and music genres to deal with the cold start problem. The ratings needed for the recommendation task were collected with our preliminary study. In this paper, we defined both a multi-dimensional contextual model, which defines the relevance order of the contextual factors, and a clustering-based predictive model for music genre recommendation. To take into account the degree of importance of the contextual factors and the relationship between music genres and contexts, we proposed an MC–CARS called MORec, that matches the user's current contextual situation and the generated clusters to rank the list of music genres that will be recommended to him/her.

The validation of the approach was performed for both predictive model and item recommendation. The experimental results back our claim up that our method outperformed the other state-of-the-art approaches, which prove the effectiveness of using MCRS in conjunction with CARS to enhance recommendation results. Experimental results showed that the K-means algorithm; we used to generate our predictive model; outperforms the PAM algorithm in terms of connectivity and stability. Besides, it was shown that MORec has better performances on precision, recall, and F-measure metrics, compared with other traditional and context-aware recommendation algorithms. In particular, a worthy of mention outcome is that MC–CARS performs better than the transformation algorithms, which try to pre-process the contextual dataset by converting it to a 2-dimensional rating matrix so that any traditional recommendation algorithms can be applied to. In addition, the adaptation algorithms worsens the performance of the recommendations, compared to our MC–CARS, by directly incorporating contexts into the prediction function.

### *Limitations and future research direction*

Despite the merits of this work, we are deeply aware that are still rooms for improvement. There are still some open challenges regarding the multi-criteria recommendation combined with contextual factors. As preliminary research, we simply considered the data we collected from 109 participants in a previous study (Ben Sassi & Ben Yahia, 2021). The extent to which the participants' musical preferences are representative of the music listeners community at large, needs further investigation. In future work, we intend to tackle it by collecting extra ratings, using our questionnaire, to obtain a larger dataset.

Although, the relation between music genres and contextual information has been uncovered by applying a clustering algorithm which groups music genres by clusters of contextual factors. Given that one of the objectives of our work is not to separate sharply music genres, we can create fuzzy clusters, so that a genre can belong to more than one cluster with different degrees of membership. For example, Blues music may belong to the time:morning cluster with a membership equals to 0.8, but also to the location:work cluster with a degree membership of 0.6.

Furthermore, the cultural background of the listener has been considered as a significant factor which can influence his/her music taste (Skowron et al., 2017). According to the authors in (Schedl et al., 2018) culture can be defined in manifold ways based on historical, political, linguistic, or religious similarities. So, analyzing cultural patterns of music consumption behavior is an important step to improve the recommendation performance. (Zangerle et al., 2020) have combined acoustic song features and culture-related features to model users' musical preferences and cultural background. The proposed culture-aware RS has a highest impact on recommendation quality. An interesting future direction is to extend our contextual model with further data utilized for capturing cultural aspects of users like indulgence and individualism. We plan to model these aspects by means of Hofstede's cultural dimensions and the World Happiness Report (Zangerle et al., 2020).

Lastly, the need for serendipity to evaluate RS has been agreed upon (Schedl et al., 2018). Serendipity measures the ability of a RS to provide relevant and surprising recommendations. Therefore, it is important to evaluate our approach by means of serendipity to avoid the over-specialization problem.

Finally, we would like to highlight that our approach can be applied by researchers and practitioners also for other related tasks like music

artists recommendation and not only for the genre and music tracks recommendation.

## CRediT authorship contribution statement

**Imen Ben Sassi:** Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing, Visualization. **Sadok Ben Yahia:** Writing - review & editing. **Innar Liiv:** Visualization.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

The authors are supported by the Astra funding program Grant 2014-2020.4.01.16-032.

## Appendix A. Comparison of Music RS

See Tables A.8 and A.9.

**Table A.8**
Comparison of surveyed works for music recommendation.

| Category | Approach | Goal | Technique | Recommended Items | Context Information | Context Relevance | Beyond Accuracy | Cold Start |
|---|---|---|---|---|---|---|---|---|
| Traditional Music RS | (Kuzelewska & Wichowski, 2015) | Songs recommendation | Clustering | Single item | – | – | No | No |
| | (Oramas et al., 2016) | Sounds and songs recommendation | Knowledge graph | Top-n items | – | – | Diversity and entropy based novelty | No |
| | (Cheng et al., 2017) | Songs recommendation | Word embedding and matrix factorization | Top-n items | – | – | No | No |
| | (Kim et al., 2018) | Songs recommendation | Matrix factorization and recurrent neural network | Playlist | – | – | No | Yes |
| | (Zheng et al., 2018) | Songs recommendation | Logistic principle component analysis, gaussian state-space, and matrix factorization | Top-n items | – | – | No | No |
| Music CARS | (Deng et al., 2015) | Songs recommendation | User-based collaborative filtering | Top-n items | Emotion | Effect of emotions' granularities | No | No |
| | (Cheng & Shen, 2016) | Songs recommendation | Support vector machine | Playlist | Location | No | No | No |
| | (Sánchez-Moreno et al., 2016) | Songs recommendation | K-nearest neighbor | Top-n items | Time | No | No | No |

**Table A.9**
Comparison of surveyed works for music recommendation (continued).

| Category | Approach | Goal | Technique | Recommended Items | Context Information | Context Relevance | Beyond Accuracy | Cold Start |
|---|---|---|---|---|---|---|---|---|
| Music CARS | (Abdul et al., 2018) | Songs recommendation | Convolutional neural networks | Top-n items | Emotion | No | No | No |
| | (Katarya & Verma, 2018) | Songs recommendation | Depth-first search & Particle swarm optimization | Top-n items | Time | No | No | No |
| | (Volokhin & Agichtein, 2018) | Songs recommendation | Random forest | Playlist | Activity | Association between activities and intents | No | No |
| | (Wang et al., 2018) | Songs recommendation | Embedding and cosine similarity | Top-n items | Local and global context | No | No | No |
| | (Andjelkovic et al., 2019) | Artists recommendation | WordNet similarity | Playlist | Mood | No | Diversity, explanation, transparency, and control | Yes |
| | (Véras et al., 2019) | Music CDs recommendation | Base cross-domain technique | Top-n items | Time, location, and companion | Information gain attribute evaluation | No | Yes |
| | (Wang et al., 2020) | Songs recommendation | Graph-based technique | Playlist | Time, location, weather, and season | No | No | Yes |
| | (Zangerle et al., 2020) | Songs recommendation | Gradient boosting decision trees | Top-n items | Cultural background | Information gain | No | No |
| | (Polignano et al., 2021) | Songs recommendation | Cosine similarity | Playlist | Emotion | No | No | Yes |

## Appendix B. Datasets description

### B.1. COMUDA: A context-aware music dataset

We collected ratings from 109 users to 22 music genres regarding 33 contextual dimension values listed in Table 1. The COMUDA dataset basic statistics are depicted in Table B.10.

In order to generalize our statistical results, we have undertaken the representation of the collected data. Clearly, the age of participants represents a pivotal factor to understand study outcomes. We give the histogram of user age in Fig. B.2. In our case, we are interested in academics, with high

**Table B.10**
COMUDA dataset statistics.

| | |
|---|---|
| Number of users | 109 |
| Number of items (music genres) | 22 |
| Number of context dimensions | 6 |
| Number of contextual dimension values | 33 |
| Number of contextual situations | 800 |
| Number of ratings | 18,668 |
| Average of users age | 25 |
| Number of men | 50 |
| Number of women | 59 |
| Maximum ratings of single user | 726 |
| Minimum ratings of single user | 2 |
| Average value of all ratings | 2.7026 |
| Standard deviation of all ratings | 1.7643 |
| Median of all rating values | 1.0000 |

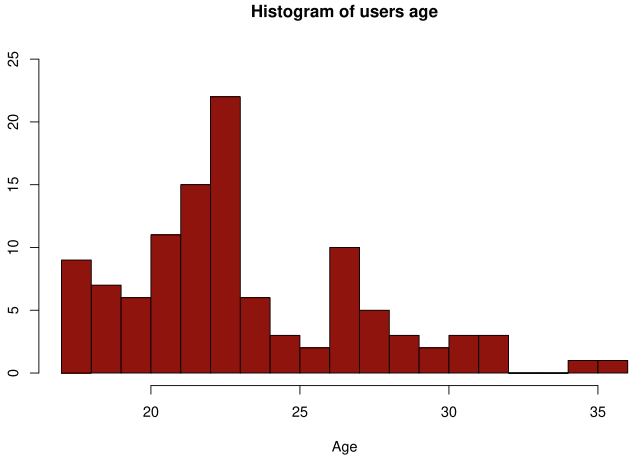



**Fig. B.2.** Histogram of users age. Most users are aged between 20 and 25, which explains the peak in this age interval.

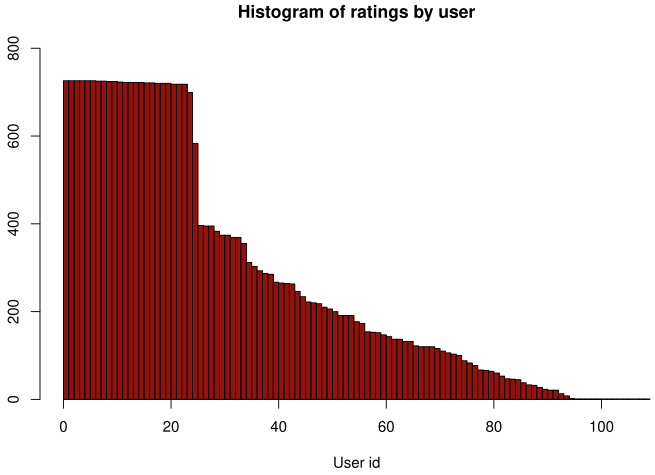



**Fig. B.3.** Histogram of the number of ratings per user. We can notice the variations of users' contribution in terms of the number of ratings, from more than 700 ratings to less than 10 ratings.

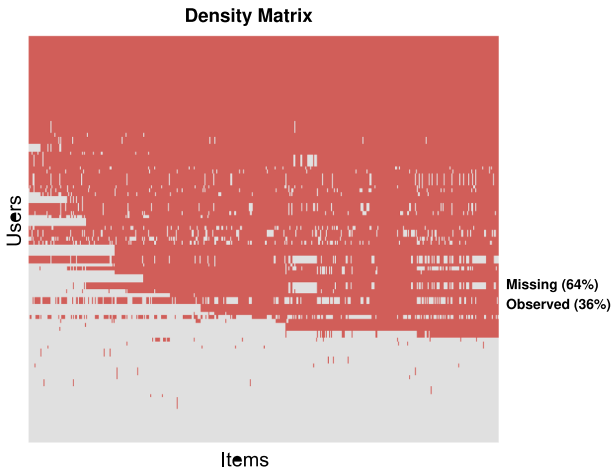



**Fig. B.4.** The density of the COMUDA dataset: Rows define users count, while columns stand for items count. Missing ratings are colored in red (64%) and given ratings with gray (36%).

computer skills, to answer the online questionnaire easily. This population choice can be noticed by the peak of the number of users aged between 20 and 25.

We detail the distribution of ratings by users in Fig. B.3. We can observe a high variability in user contribution, from those who provided fewer than 10 ratings to many users with more than 700 ratings.

We present the structure and density of the COMUDA dataset in Fig. B.4. We computed the density of a dataset through the users' given ratings and the total number of items. In our case, the assigned ratings are shown in gray, users and items are reordered with a seriation method (Hahsler, Hornik, & Buchta, 2008; Liiv, 2010) to enable visual consistency with the histogram on Fig. B.3 and better visual representation of underlying structure and patterns. The sparsity level, defined by the ratio of observed to total ratings, affects different recommendation methods in different ways. Some techniques perform better in sparse datasets while other ones in dense settings. Our dataset represents a density level equal to 64%.

### *B.2. The MusiClef dataset: MIR:MusiClef:2012: MMSys: version1.0*

The MusiClef dataset contains five music content features, namely editorial metadata (artists, albums, songs, and MusicBrainz identifiers), audio features, web pages, collaborative tags, and professional tags. Music items were tagged using two methods. First, we gathered collaborative tags from Last.fm to automatically tag items with weighted tags (normalized between [0,100]). Second, music tracks were tagged manually by experts (a group of professional music consultants) using 94 different tags describing songs genre and mood, from 355 tags, i.e., 167 genres and 188 moods. $8,944$ <song-id, tag> entries, in total, were collected and were divided into $6,473$ assignments for training and $2,471$ for tests. For each song, the dataset contains at least one tag for the genre and five tags for mood. For our task, we are only interested in genre annotations. Thus, we have retrieved $1,355$ <song-id, tag> entries where tag represents a genre tag. These tags define music sub-genres. For example, professional annotators have distinguished between free jazz, nu-jazz, cool jazz, and big band, as far as they tag a jazz song. We have enriched the retrieved <song-id, tag> entries with 48 new entries annotated with the music genres absent in the MusiClef dataset, i.e., Holiday, Children's music, Soundtracks, Religious, Alternative, and Latino. As a result, we have gathered a total of $1,403$ entries of labeled songs as <song, artist, album, tag>.

### *B.3. The diary test dataset*

Table B.11 shows some statistics about the test dataset including the user, his/her age, his/her gender, the number of ratings given to music genres in different contextual situations, and considering no contextual information, as well the average of his/her ratings. Table B.11 shows that 9 of the participants were female (81.8%) and 2 were male (18.2%). The age of participants varied between 21 and 29, with a median equal to 24 and a mean of 24.64. We asked participants to provide no-context ratings, which define their preferred music genres using a scale from 1 to 5 regardless of any context factor into account, and in-context rating data that express their preferences for music genres in each contextual situation.

**Table B.11**
Test Diary dataset statistics.

| User | Age | Gender | Number of in-context ratings | Number of no-context ratings | Average of ratings |
|---|---|---|---|---|---|
| User 1 | 24 | Female | 23 | 22 | 1 |
| User 2 | 23 | Male | 45 | 22 | 3 |
| User 3 | 25 | Female | 721 | 17 | 1 |
| User 4 | 21 | Female | 38 | 1 | 1 |
| User 5 | 29 | Male | 263 | 22 | 3 |
| User 6 | 23 | Female | 722 | 22 | 2 |
| User 7 | 23 | Female | 106 | 22 | 3 |
| User 8 | 27 | Female | 718 | 19 | 3 |
| User 9 | 27 | Female | 718 | 19 | 3 |
| User 10 | 26 | Female | 724 | 22 | 3 |
| User 11 | 23 | Female | 725 | 22 | 1 |

## Appendix C. Clustering evaluation

We have executed the K-means algorithm with different values of $K = 3,6,10,15,22,23,24,25$. Table C.12 shows the internal validation indices we used in our experiments (see (Desgraupes, 2018) for more details). We have selected the most used measures of the literature. The rule column shows the condition in which we consider the clustering results as a good partition based on a validation index. In other terms, *max* means that higher index values show better clustering results. However, *min* means that lower index values show better clustering schema.

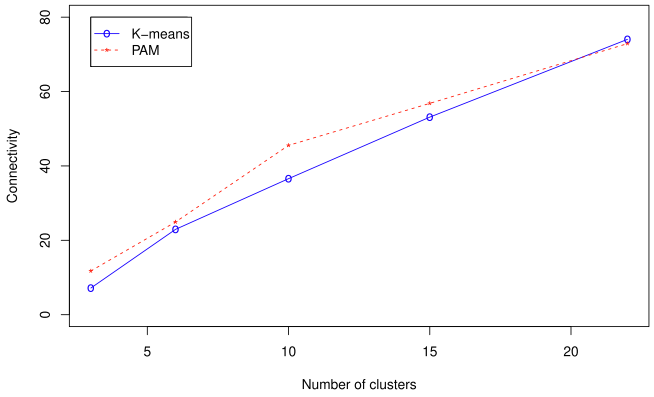
As depicted in Table C.13, most of the best results (marked in bold) given by the 17 selected internal indices, are obtained with 22 clusters, which represents the number of music genres in our scenario. Indeed, we obtained the best results with $K = 22$ for 13 validation indices in terms of maximization or minimization of the measure (c.f., the last column in Table C.12). For example, the Total Variance, denoted TV, shows the separation between clusters and represents a good mean to identify the number of the optimum clusters. The total variance is computed as $TV = \frac{between-SS}{total-SS}$, where $between-SS$ (between Sum of Squares) refers to the separation between clusters, whereas $total-SS$ (total Sum of Squares) refers to the sum of "within-Sum" of Squares and "between-Sum" of Squares. The K-means algorithm minimizes the "intra-group" dispersion and maximizes the "inter-group" dispersion. As shown in Table C.13, as far as the number of clusters increases, the TV index increases and reaches its best result for $K = 22$. Loosely speaking, the TV index has increased from 0.810425 to reach 0.991686 for $K = 22$ and then decreased to 0.980000. Besides, for the Dunn index, denoted Du, as far as the number of clusters increases, the Du index increases, i.e., the computed value has improved from 0.062601 to 0.792900 when augmenting the number of partitions from $K = 3$ to $K = 22$ with a significant improvement of 0.7303. Then, it decreased when $K$ exceeds 22 to reach 0.219167 with $K = 25$. These results prove we obtained the most separated and compact clusters with $K = 22$. However, for the other evaluation indices, i.e., Ha, RL, RT, and Sil (c.f., Table C.12), the obtained values are not bad even if they are not the optimal ones. For example, the silhouette index, denoted Sil that measures the similarity of an object to its own cluster compared to other ones, has values interval is between $-1$ and 1, and our obtained value with $K = 22$ is 0.202754. These results provide evidence of the high similarity between inter-cluster objects, that represent contextual situations in which the music genre (presented by the cluster) is preferred.

**Table C.12**
Internal cluster validation indices.

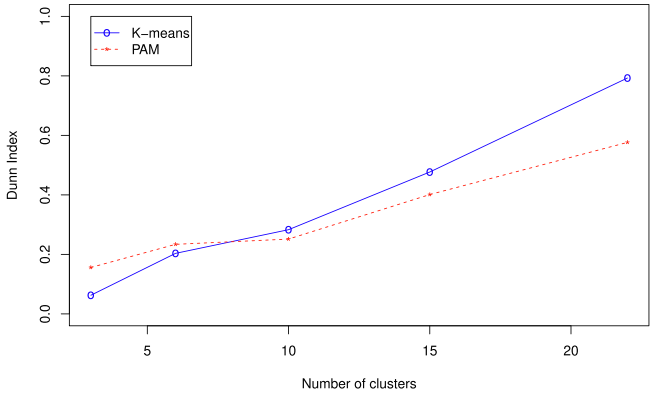
| Index | Notation | Rule |
|---|---|---|
| C-index | Ci | min |
| Calinski-Harabasz | CH | max |
| Davies-Bouldin | DB | min |
| Dunn | Du | max |
| Gamma | Ga | max |
| G+ | G+ | min |
| Hartigan | Ha | min |
| McClain-Rao | McCR | min |
| PBM | PMB | max |
| Point biserial | Pb | max |
| Ratkowsky-Lance | RL | max |
| Ray-Turi | RT | min |
| SD Scat | SDS | min |
| Silhouette average | Sil | max |
| Wemmert-Gancarski | WG | max |
| Xie-Beni | XB | min |
| Total variance | TV | max |

**Table C.13**
Internal cluster validation measures results, with a variation of the number of clusters, using the COMUDA dataset. Bold values show optimal values for all indices.

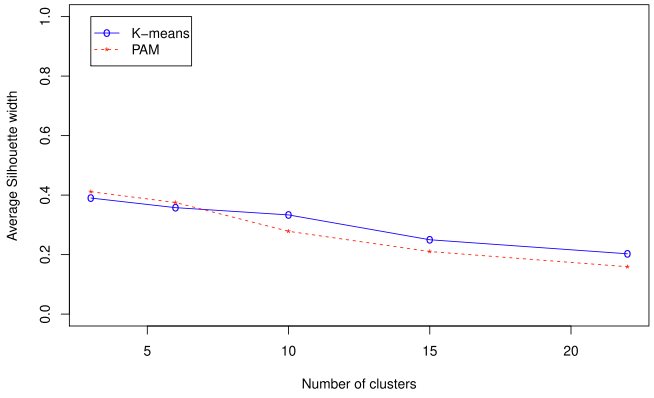
| Index | 3-clusters | 6-clusters | 10-clusters | 15-clusters | 22-clusters | 23-clusters | 24-clusters | 25-clusters |
|---|---|---|---|---|---|---|---|---|
| Ci | 0.056691 | 0.071717 | 0.039291 | 0.048330 | **0.013409** | 0.014973 | 0.054379 | 0.025189 |
| CH | 64.124320 | 29.322820 | 55.725870 | 35.958310 | **74.722450** | 70.803990 | 22.751410 | 45.466050 |
| DB | 0.791197 | 0.798285 | 0.620311 | 0.204466 | **0.084953** | 0.246413 | 0.812026 | 0.389508 |
| Du | 0.062601 | 0.203500 | 0.283000 | 0.476900 | **0.792900** | 0.154347 | 0.099151 | 0.219167 |
| Ga | 0.747004 | 0.696956 | 0.744314 | 0.674208 | **0.923103** | 0.918844 | 0.604009 | 0.821635 |
| G+ | 0.060096 | 0.042932 | 0.025049 | 0.019665 | **0.002401** | 0.003099 | 0.008093 | 0.002667 |
| Ha | **1.452774** | 1.691967 | 3.082175 | 3.331046 | 4.960408 | 5.048373 | 4.062897 | 4.915578 |
| McCR | 0.285169 | 0.333017 | 0.255775 | 0.283360 | **0.162645** | 0.167033 | 0.303128 | 0.190590 |
| PBM | 122.461900 | 68.344340 | 134.616200 | 99.891940 | **241.394600** | 177.204200 | 121.387300 | 153.243400 |
| Pb | −3.068158 | −1.805522 | −1.617045 | −1.176624 | **−0.969234** | −1.883871 | −1.644409 | −1.638688 |
| RL | **0.525277** | 0.378657 | 0.309367 | 0.253872 | 0.212394 | 0.207834 | 0.202510 | 0.199288 |
| RT | **0.632985** | 2.846302 | 1.178723 | 8.322029 | 1.678146 | 1.537736 | 4.076072 | 0.671642 |
| SDS | 0.409903 | 0.209551 | 0.044753 | 0.023385 | **0.007008** | 0.007889 | 0.012542 | 0.008452 |
| Sil | **0.390133** | 0.357800 | 0.333400 | 0.249900 | 0.202754 | 0.150800 | 0.134400 | 0.130900 |
| WG | 0.567510 | 0.515872 | 0.549551 | 0.638134 | **0.737356** | 0.703466 | 0.697677 | 0.618627 |
| XB | 8.222368 | 8.542207 | 2.504358 | 8.322029 | **1.678146** | 1.837736 | 4.076072 | 1.870167 |
| TV | 0.810425 | 0.836787 | 0.956103 | 0.984195 | **0.991686** | 0.984000 | 0.983000 | 0.980000 |




(a)




(b)




c

**Fig. C.5.** Comparison between the K-means and the PAM algorithm based on internal criteria: (a) based on the Connectivity; (b) based on the Dunn index; (c) based on the Average Silhouette width.

We also compared K-means with the PAM algorithm (Kaufman & Rousseeuw, 1990), for $K = 3, 6, 10, 15, 22$ clusters, based on three internal criteria (c.f., Fig. C.5) including Connectivity measure, Silhouette average and Dunn index (see the *category:internal* of the clValid package in (Brock, 2014) for further details).

Fig. C.5(a) shows a comparison between K-means and PAM in terms of the Connectivity criteria. The latter shows the degree of connectedness of the clusters, which has to be minimized. Our findings show that, for $K = 3, 6, 10, 15$, the PAM algorithm gives the optimal scores in terms of Connectivity. However, with $K = 22$, K-means outperforms PAM. Fig. C.5(b) depicts the Dunn index evaluation of K-means and PAM for different $K$

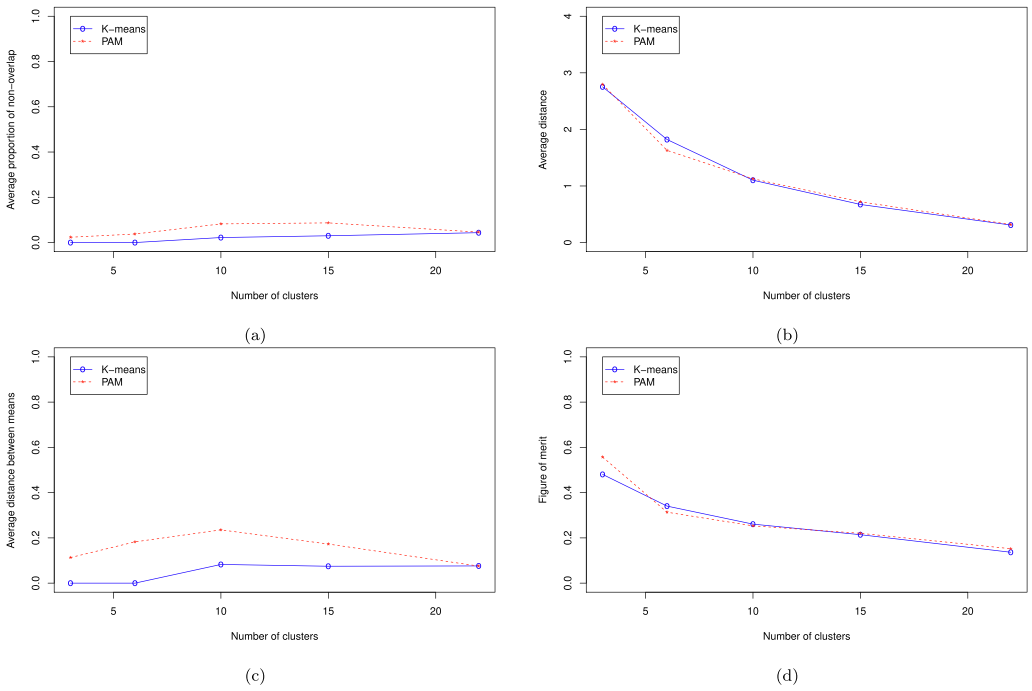



**Fig. C.6.** Comparison between the K-means and the PAM algorithm in terms of stability: (a) based on the Average Proportion of Non-overlap (APN); (b) based on the Average Distance (AD); (c) based on the Average Distance between Means (ADM); (d) based on the Figure Of Merit (FOM).

values. This measure is computed by the ratio between the smallest distance between objects not in the same cluster and the largest intra-cluster distance. The latter distance has to be maximized. Results in Fig. C.5(b) show that the Dunn index values got with K-means are almost better than those yielded by the PAM algorithm. Fig. C.5(c) presents the Average Silhouette width that measures the separation of the clusters. The values of silhouette width range from $-1$ (for poorly clustering) to 1 (for well-clustered data). Fig. C.5(c) shows that starting from 10 clusters, as far as the number of clusters increases, the Average Silhouette width of the K-means algorithm is better performing than the PAM algorithm.

Results related to stability comparison show that compared to the PAM algorithm, the K-means is more stable (c.f., Fig. C.6). Indeed, whatever the selected number of clusters, K-means gives better stability values. We used for stability evaluation four different stability measures, from the *category: stability* of the clValid package (Brock, 2014), including the average proportion of non-overlap (c.f., Fig. C.6(a)), the average distance (c.f., Fig. C.6(b)), the average distance between means (c.f., Fig. C.6(c)), and the figure of merit (c.f., Fig. C.6(d)). These metrics assess the stability of the resulted clusters by comparing them with the clusters obtained by removing one column at a time. For the four measures, the cluster with smaller values is considered as the one with better clustering results. As depicted in Fig. C.6, the curve of the K-means algorithm (c.f., the continuous blue line) is often below the one generated with the PAM algorithm (c.f., the dashed red line).

## References


Abdul, A., Chen, J., Liao, H. Y., & Chang, S. H. (2018). An emotion-aware personalized music recommendation system using a convolutional neural networks approach. *Applied Sciences, 8*. https://doi.org/10.3390/app8071103

Adomavicius, G., & Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering, 24*, 896–911.

Adomavicius, G., & Kwon, Y. (2015). Multi-criteria recommender systems. In *Recommender systems handbook* (2nd ed., pp. 847–880). Springer.

Adomavicius, G., Manouselis, N., & Kwon, Y. (2011). Multi-criteria recommender systems. In *Recommender systems handbook* (pp. 769–803). Springer.

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems, 23*, 103–145.

Adomavicius, G., & Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook* (pp. 217–253). Springer.

Andjelkovic, I., Parra, D., & O'Donovan, J. (2019). Moodplay: Interactive music recommendation based on artists' mood similarity. *International Journal of Human-Computer Studies, 121*, 142–159. URL: http://www.sciencedirect.com/science/article/pii/S1071581918301654, doi: 10.1016/j.ijhcs.2018.04.004. advances in Computer-Human Interaction for Recommender Systems.

Ayata, D., Yaslan, Y., & Kamasak, M. E. (2018). Emotion based music recommendation system using wearable physiological sensors. *IEEE Transactions on Consumer Electronics, 64*, 196–203. https://doi.org/10.1109/TCE.2018.2844736

Bai, K., & Kawagoe, K. (2018). Background music recommendation system based on user's heart rate and elapsed time. In *Proceedings of the 2018 10th international conference on computer and automation engineering* (pp. 49–52). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3192975.3193013.

Baltrunas, L., Ludwig, B., Peer, S., & Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing, 16*, 507–526.

Ben Sassi, I., & Ben Yahia, S. (2021). How does context influence music preferences: a user-based study of the effects of contextual information on users' preferred music. *Multimedia Systems, 27*, 143–160. URL: https://link.springer.com/article/10.1007/s00530-020-00717-x, doi: 10.1007/s00530-020-00717-x.

Ben Sassi, I., Ben Yahia, S., & Mellouli, S. (2017a). Context-aware recommender systems in mobile environment: On the road of future research. *Information Systems, 72*, 27–61.

Ben Sassi, I., Ben Yahia, S., & Mellouli, S. (2017b). User-based context modeling for music recommender systems. In *Proceedings of the 23rd international symposium on methodologies for intelligent systems – foundations of intelligent systems* (pp. 157–167). Warsaw, Poland: Springer.

Bonnin, G., & Jannach, D. (2014). Automated generation of music playlists: Survey and experiments. *ACM Computing Survey, 47*. https://doi.org/10.1145/2652481

Braunhofer, M., Elahi, M., Ge, M., Ricci, F., & Schievenin, T. (2013). STS: Design of weather-aware mobile recommender systems in tourism. In *Proceedings of the AI*IA Intl. Workshop on Intelligent User Interfaces, Turin, Italy* (pp. 40–46).

Brock, G., Pihur, V., Datta, S., & Datta, S. (2014). *clValid, an R package for cluster validation. r package version 0.6-6 ed.* Louisville, Kentucky, U.S: Department of Bioinformatics and Biostatistics, University of Louisville.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785.

Cheng, Z., & Shen, J. (2016). On effective location-aware music recommendation. *ACM Transactions on Information Systems, 34*. https://doi.org/10.1145/2846092

Cheng, Z., Shen, J., Zhu, L., Kankanhalli, M., & Nie, L. (2017). Exploiting music play sequence for music recommendation. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 3654–3660). AAAI Press.

Deng, S., Wang, D., Li, X., & Xu, G. (2015). Exploring user emotion in microblogs for music recommendation. *Expert Systems with Applications, 42*, 9284–9293. URL: https://www.sciencedirect.com/science/article/pii/S0957417415005746, doi: 10.1016/j.eswa.2015.08.029.

Desgraupes, B. (2018). clusterCrit: Clustering Indices. r package version 1.2.8 ed. University Paris Ouest. Paris, France.

Dey, A. K., & Abowd, G. D. (1999). Towards a better understanding of context and context-awareness. In H. W. Gellersen (Ed.), *Proceedings of the first international symposium handheld and ubiquitous computing* (pp. 304–315). London, UK: Springer Berlin Heidelberg.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis*. New York, NY, USA: Wiley-Interscience.

Ekman, P. (1999). Basic emotions. In *Handbook of cognition and emotion* (pp. 45–60). John Wiley.

Hahsler, M., Hornik, K., & Buchta, C. (2008). Getting things in order: An introduction to the r package seriation. *Journal of Statistical Software, 25*, 1–34.

Haim, M., Andreas, G., & Brosius, H. B. (2018). Burst of the filter bubble? Effects of personalization on the diversity of google news. *Digital Journalism, 6*, 330–343. https://doi.org/10.1080/21670811.2017.1338145

Hartigan, J. A., & Wong, M. A. (1979). A k-means clustering algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics), 28*, 100–108.

Hasan, S., Zhan, X., & Ukkusuri, S. V. (2013). Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the international workshop on urban comput.* (pp. 1–8). ACM, New York, NY, USA.

Hong, M., & Jung, J. J. (2021). Multi-criteria tensor model for tourism recommender systems. *Expert Systems with Applications, 170*, 114537. URL: https://www.sciencedirect.com/science/article/pii/S0957417420311817, doi: 10.1016/j.eswa.2020.114537.

Ignatov, D. I., Nikolenko, S. I., Abaev, T., & Poelmans, J. (2016). Online recommender system for radio station hosting based on information fusion and adaptive tag-aware profiling. *Expert Systems with Applications, 55*, 546–558. URL: https://www.sciencedirect.com/science/article/pii/S0957417416300513, doi: 10.1016/j.eswa.2016.02.020.

Jannach, D., Karakaya, Z., & Gedikli, F. (2012). Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM conference on electronic commerce, association for computing machinery* (pp. 674–689). New York, NY, USA. URL: doi: 10.1145/2229012.2229065.

Jiang, S., Ferreira, J., & González, M. C. (2012). Clustering daily patterns of human activities in the city. *Data Mining and Knowledge Discovery, 25*, 478–510.

Kaminskas, M., & Ricci, F. (2012). Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review, 6*, 89–119.

Katarya, R., & Verma, O. P. (2018). Efficient music recommender system using context graph and particle swarm. *Multimedia Tools Applications, 77*, 2673–2687. https://doi.org/10.1007/s11042-017-4447-x

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. New York, NY, USA: Wiley.

Kim, J., Won, M., Liem, C. C. S., & Hanjalic, A. (2018). Towards seed-free music playlist generation: Enhancing collaborative filtering with playlist title information. In *Proceedings of the ACM recommender systems challenge 2018* (pp. 1–6). Association for Computing Machinery, New York, NY, USA. doi: 10.1145/3267471.3267485.

Kuzelewska, U., & Wichowski, K. (2015). A modified clustering algorithm dbscan used in a collaborative filtering recommender system for music recommendation. In *Proceedings of the international conference on dependability and complex systems* (pp. 245–254). LwÓwek Ślaski, Poland: Springer International Publishing.

Lee, J. H. (2011). How similar is too similar? Exploring users perception of similarity in playlist evaluation. In *Proceedings of the 12th international conference on music information retrieval, Miami, FL, USA* (pp. 109–114).

Liiv, I. (2010). Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 3*, 70–91.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology, 22*, 1–55.

Macefield, R. (2009). How to specify the participant group size for usability studies: A practitioner's guide. *Journal of Usability Studies, 5*, 34–45.

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

Ono, C., Takishima, Y., Motomura, Y., & Asoh, H. (2009). Context-aware preference model based on a study of difference between real and supposed situation data. In *Proceedings of the 17th international conference on user modeling, adaptation, and personalization: Formerly UM and AH* (pp. 102–113). Berlin, Heidelberg: Springer-Verlag.

Oramas, S., Ostuni, V. C., Noia, T. D., Serra, X., & Sciascio, E. D. (2016). Sound and music recommendation with knowledge graphs. *ACM Transactions on Intelligent Systems and Technology, 8*. https://doi.org/10.1145/2926718

Polignano, M., Narducci, F., de Gemmis, M., & Semeraro, G. (2021). Towards emotion-aware recommender systems: An affective coherence model based on emotion-driven behaviors. *Expert Systems with Applications, 170*, 114382. URL: https://www.sciencedirect.com/science/article/pii/S0957417420310575, doi: 10.1016/j.eswa.2020.114382.

Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of Computers and Communications, 5*, 27–34.

Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender systems handbook* (2nd Ed.). Springer Publishing Company, Incorporated.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2010). *Recommender systems handbook* (1st Ed.). Berlin, Heidelberg: Springer-Verlag.

Sánchez-Moreno, D., Gil González, A. B., Muñoz Vicente, M. D., López Batista, V. F., & Moreno García, M. N. (2016). A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Systems with Applications, 66*, 234–244. URL: https://www.sciencedirect.com/science/article/pii/S0957417416304973, doi: 10.1016/j.eswa.2016.09.019.

Sánchez-Moreno, D., Zheng, Y., & Moreno-García, M. N. (2018). Incorporating time dynamics and implicit feedback into music recommender systems. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence (WI)* (pp. 580–585). https://doi.org/10.1109/WI.2018.00-34

Schedl, M., & Ferwerda, B. (2017). Large-scale analysis of group-specific music genre taste from collaborative tags. In *Proceedings of the 19th IEEE international symposium on multimedia* (pp. 479–482). IEEE Computer Society. https://doi.org/10.1109/ISM.2017.95.

Schedl, M., Liem, C. C., Peeters, G., & Orio, N. (2013). A professionally annotated and enriched multimodal data set on popular music. In *Proceedings of the 4th ACM multimedia systems conference, Oslo, Norway* (pp. 78–83).

Schedl, M., Zamani, H., Chen, C., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval, 7*, 95–116.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender Systems Handbook* (pp. 257–297). Springer.

Skowron, M., Lemmerich, F., Ferwerda, B., & Schedl, M. (2017). Predicting genre preferences from cultural and socio-economic factors for music retrieval. In *Proceedings of the 39th European conference on advances in information retrieval* (pp. 561–567). https://doi.org/10.1007/978-3-319-56608-5_49

Srivastava, R., Hingmire, S., Palshikar, G. K., Chaurasia, S., & Dixit, A. (2016). Csrs: A context and sequence aware recommendation system. In *Proceedings of the 8th annual meeting of the forum on information retrieval evaluation* (pp. 8–15). New York, NY, USA: ACM.

Telgarsky, M., & Vattani, A. (2010). Hartigan's method: k-means clustering without voronoi. *Journal of Machine Learning Research – Proceedings Track, 9*, 820–827.

Van Eeuwijk, P., & Angehrn, Z. (2017). How to...Conduct a Focus Group Discussion (FGD). Methodological Manual. swiss tph - fact sheet society, culture and health ed. University of Basel. Basel.

Vargas, S., Baltrunas, L., Karatzoglou, A., & Castells, P. (2014). Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proceedings of the 8th ACM conference on recommender systems* (pp. 209–216). New York, NY, USA: ACM.

Véras, D., Prudêncio, R., & Ferraz, C. (2019). Cd-cars: Cross-domain context-aware recommender systems. *Expert Systems with Applications, 135*, 388–409. URL: https://www.sciencedirect.com/science/article/pii/S095741741930421X, doi: 10.1016/j.eswa.2019.06.020.

Volokhin, S., & Agichtein, E. (2018). Towards intent-aware contextual music recommendation: Initial experiments. In *Proceedings of the 41st International ACM SIGIR conference on research & development in information retrieval* (pp. 1045–1048). https://doi.org/10.1145/3209978.3210154

Volokhin, S., & Agichtein, E. (2018). Understanding music listening intents during daily activities with implications for contextual music recommendation. In *Proceedings of the 2018 conference on human information interaction & retrieval* (pp. 313–316). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3176349.3176885.

Wang, D., Deng, S., Zhang, X., & Xu, G. (2018). Learning music embedding with metadata for context aware recommendation. *World Wide Web, 1399–1423*. https://doi.org/10.1007/s11280-017-0521-6

Wang, R., Ma, X., Jiang, C., Ye, Y., & Zhang, Y. (2020). Heterogeneous information network-based music recommendation system in mobile networks. *Computer Communications, 150*, 429–437. URL: http://www.sciencedirect.com/science/article/pii/S0140366419311399, doi: 10.1016/j.comcom.2019.12.002.

Zangerle, E., Pichl, M., & Schedl, M. (2020). User models for culture-aware music recommendation: Fusing acoustic and cultural cues. *Transactions of the International Society for Music Information Retrieval, 3*, 1–16. https://doi.org/10.5334/tismir.37

Zhang, K., Liu, X., Wang, W., & Li, J. (2021). Multi-criteria recommender system based on social relationships and criteria preferences. *Expert Systems with Applications*, 114868. URL: https://www.sciencedirect.com/science/article/pii/S0957417421003092, doi: https://doi.org/10.1016/j.eswa.2021.114868.

Zheng, E., Kondo, G. Y., Zilora, S., & Yu, Q. (2018). Tag-aware dynamic music recommendation. *Expert Systems with Applications, 106*, 244–251. URL: https://www.sciencedirect.com/science/article/pii/S0957417418302446, doi: 10.1016/j.eswa.2018.04.014.

Zheng, Y. (2017). Situation-aware multi-criteria recommender system: Using criteria preferences as contexts. In *Proceedings of the symposium on applied computing* (pp. 689–692). ACM, New York, NY, USA.

Zheng, Y., & Jose, A. A. (2019). Context-aware recommendations via sequential predictions. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 2525–2528). Association for Computing Machinery, New York, NY, USA. doi: 10.1145/3297280.3297639.

Zheng, Y., Mobasher, B., & Burke, R. (2015). Carskit: A java-based context-aware recommendation engine. In *Proceedings of the 2015 IEEE international conference on data mining workshop (ICDMW)* (pp. 1668–1671). USA: IEEE Computer Society. https://doi.org/10.1109/ICDMW.2015.222.

Zheng, Y., Shekhar, S., Jose, A. A., & Rai, S. K. (2019). Integrating context-awareness and multi-criteria decision making in educational learning. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 2453–2460). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3297280.3297522.