# A User-centered Music Recommendation Approach for Daily Activities

Ricardo Dias    Ricardo Cunha
INESC-ID, Instituto Superior Técnico,
Universidade de Lisboa
Lisboa, Portugal
{ricardo.dias,
ricardo.d.cunha}@tecnico.ulisboa.pt

Manuel J. Fonseca
Faculdade de Ciências,
Universidade de Lisboa
Lisboa, Portugal
mjf@di.fc.ul.pt

## ABSTRACT
The number of songs available on the Internet has grown steadily over the last decade, with the recent growth being due mainly to streaming services. As a consequence, it is extremely difficult for users to find the appropriate music that suit their needs, in particular, while using systems that do not have any previous information about them. This is further exacerbated while selecting appropriate songs for daily activities, like shopping, running or sleeping. In this paper we describe Improvise, a personalized music recommendation solution for daily activities, whose approach associates music content (acoustic features) with activities (context). Each activity is characterized by determining intervals for each content feature, which are then used to filter out songs to be suggested to users. While the initial intervals are generic enough to provide recommendations for different activities without having previous knowledge about the user's tastes, our approach also considers users' feedback to personalize the recommendations for each user and activity. This is done by adapting the intervals according to the feedback from users. Preliminary evaluation shows that we are on the good path to achieve the goal of developing a solution to effectively recommend songs for daily activities, and able to adjust to individual user's tastes, increasing their satisfaction.

## Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Retrieval Models

## General Terms
Design, Algorithms, Human Factors, Experimentation

## Keywords
User-centered, Music Recommendation, Content, Context, Daily Activities, Cold-Start


*CBRecSys 2014*, October 6, 2014, Silicon Valley, CA, USA.

## 1. INTRODUCTION
Over the last decade, due to the increasingly easy access to online music streaming services, people have gained the opportunity to listen to millions of songs over the Internet and almost everywhere. However, the opening to this broad spectrum of songs lead people to feel paralyzed and doubtful [17], as it gets harder for them to filter out the songs they would enjoy the most, specially while performing other activities, such as practicing sports, driving or studying [11]. Due to the large number of songs that users have access, it is hard for them to select the most appropriate songs for their activities.

To reduce the burden of choosing among too many songs, researchers have focused on creating recommender systems that can automatically generate recommendations that fit users' preferences. Celma's extensive work on recommendation [5] classifies recommender systems into five typical categories: *Demographic Filtering, Collaborative Filtering, Context Aware Filtering, Content-Based Filtering* and *Hybrid Methods*. Recently, hybrid and context-aware approaches have gained relevance amongst researchers, as they agree that listening patterns can be influenced by different factors, such as temporal properties [7], location, emotions and the activity a listener is engaged in [21].

Concerning music selection for daily activities, several approaches have been proposed. In [21], the authors created a mobile system that is able to detect what activity the user is performing and select the appropriate music for it, based on the time of the day, accelerometer data, and audio from the the microphone. Lifetrak [15] is a context-aware playlist generator that automatically chooses music in real-time based upon the location, the pace of movement, the current time, and other phenomena in the users environment. It uses a simple learning mechanism to adjust the ratings of songs for a particular context based on users feedback when a song is being played. However, these approaches are still very impersonal with little control, lacking users involvement through the whole steps of the recommendation process, which could improve the users satisfaction and confidence [9, 20].

In this work we describe Improvise, a user-centered recommendation approach for daily activities. Improvise is a recommendation model developed by characterizing activities in terms of content features, by defining their boundaries.

The set of intervals generated for each feature and activity are then used to filter out songs to be suggested. We developed a generic model with this approach (to be used initially by all users) by gathering data from several users. Individual personalization of this model is created over time by taking into account user's feedback for each activity, and consequently adapting the intervals based on the features from the new selected songs.

Preliminary experimental evaluation revealed that the initial generic model was able to suggest songs to daily activities for different users, without any knowledge about them. Moreover, by using the user feedback (selected songs from the recommendation list) to personalize the recommendation model, increased the number of adequate songs for each activity in 25% (while compared to the generic model) and the users satisfaction.

We highlight as main contributions the following: i) a user-centered personalized solution to perform recommendations for daily activities; ii) an alternative approach for solving the cold-start problem.

In the next section we discuss the related work, presenting research made in music recommender systems. Section 3 describes the proposed solution for music recommendation, and in Section 4 we present the experimental user evaluation and discuss the preliminary results of our work. Finally, in Section 5 we present future work and conclude this research.

## 2. RELATED WORK

Several approaches for music recommendation have been developed so far. Depending on the type of data used to perform the recommendations, we can categorize the methods in different groups [5], either if they use demographic data [22], listening habits and ratings [10, 6], content information from songs [4], the context when they were listened [12, 19], or any combination of the previous [18, 16]. Although collaborative filtering solutions have been the most widely researched techniques in the past, nowadays, a huge effort has been put on techniques that capture the context around the listening activity [16], because they can provide insights about the reasons that lead users to listen to certain songs.

Content-based approaches use the description of songs to compute similarities and recommend songs similar to the user favorite ones or to chosen seeds. These approaches solve the *early-rater* and popularity bias problems, as all the items are considered to be of equal importance [5] (without human intervention). However, a potential problem of these approaches is the *novelty* problem. Assuming that the *similarity function* works accurately, one might assume that a user will always receive items too similar to the ones in his profile. To cope with this problem, recommenders should use other factors to promote the *diversity* and *novelty* of the recommended items. In the solution proposed by Cano [4], acoustic features of songs (*timbre, meter and rhythm patterns*) were used for recommendation. Daniel M. [13] used *lyric feature analysis* to find similar items that describe *race conflicts* and *social issues.* In [3], Cai recommends music based only on emotion.

Context can be defined as any information that can be used to characterize the listening process [1], such as, the place where we are listening to the music, the time of day, the activity we are performing, etc. Context aware recommendation systems (CARS) use *context information* to describe and characterize the songs or artists we listen to. For example, Su et al. [19] improved Collaborative Filtering (CF) methods combining user grouping by location, motion, calendar, environment conditions and health conditions, while using content analysis to assist the system in the selection of the appropriate songs. On the other hand, Park [14] developed a modified User-based CF method (called Session-based CF), where users were replaced by sessions, adding a temporal dimension to CF recommendations. In [12], Liu et al. took the change in the interests of users over time into consideration and added time scheduling to the music playlist. Baltrunas et al. [2] introduced a new context-aware recommendation approach called user micro-profiling, where the user profile is split into several sub-profiles, each one representing the user in a particular context. The authors stated that the choice of songs during the day is influenced by contextual conditions, such as, the time of day, mood or the current *activity* listeners perform. In [23], the authors presented a novel and improved statistical model for characterizing user preferences in consuming social media content. By taking into account information about listening sessions of individual users, they have arrived at a new session-based hierarchical graphical model that enhanced individual user experience.

In short, there have been some effort from researchers to create automatic mechanisms that characterize users preferences through the use of different sorts of data, like temporal patterns, emotions, or choices behind song selection for particular activities. On the other hand, content features have been extensively used for recommendation and playlist generation because of the benefits they present. Despite that, there has been little work on engaging users through the recommendation process, giving them the control over how profiles are created and managed. We intend to tackle this gap by developing a recommendation approach based on user input and feedback as well as on content features, for creating a customizable recommendation model for each user.

## 3. IMPROVISE

In this research we describe Improvise, a personalized recommendation system able to suggest songs that fit the users needs while performing daily activities. To achieve this goal we followed a user-centered approach, taking advantage of users' input and feedback to develop a generic model capable of recommending songs to everyone, even without any previous knowledge about them. Activities were characterized using content-based features. Five activities were considered based on the existing related work [21]: *walking, relaxing, running, sleeping and shopping.*

### 3.1 Approach Overview

Figure 1 shows the different steps for creating the recommendation models, explaining the recommendation process. First, we associate songs with activities by using the user input gathered through a web-application (Figure 1-1). This allowed us to analyze what songs were more suitable for each activity (based on the users preferences) and thereby create

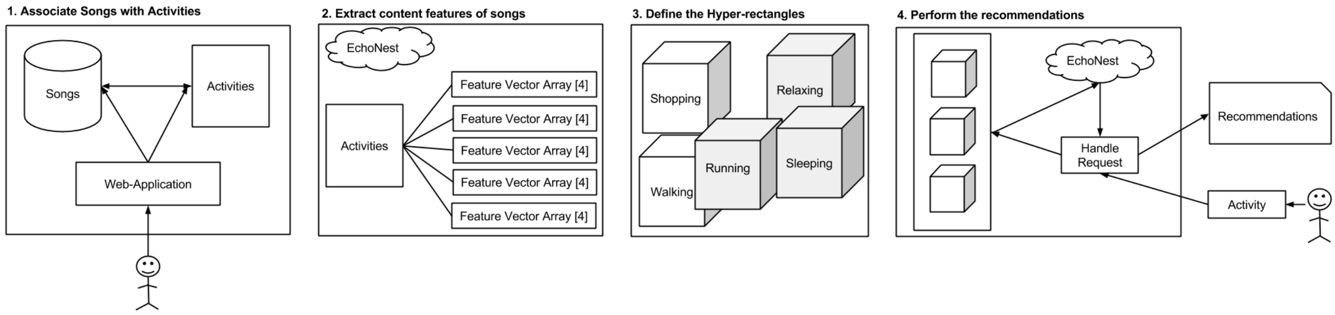



Figure 1: The different steps in Improvise to generate recommendations.

the broad and generic recommendation model. To this end, we then gathered content features from the songs, using the *EchoNest*[1] service (Figure 1-2). The values for each feature were collected to characterize each activity, by inferring intervals of values for each feature. These intervals define a set of *hyper-rectangles* (Figure 1-3), which Improvise uses to generate recommendations by filtering out songs from the *EchoNest* service (Figure 1-4).

To personalize the recommendation model we consider the songs selected by the users as appropriate for each activity (user's feedback), extract the content properties for these new songs and recalculate the hyper-rectangles for each activity, repeating steps 2 and 3. By using this approach our solution adapts the recommendations to the users' tastes and preferences over time.

In the following sections we provide details for the different steps described here.

## 3.2 Association between songs and activities

To develop a music recommendation system able to suggest songs suitable for different activities is necessary to capture the users' tastes and preferences for those activities. For example, to understand the reasons *"why do users select certain songs for running, and others for relaxing?"*. Although different criteria can influence this selection, some conceptual properties are shared among users for the same activities, such as, familiarity or distraction. However, these sort of more subjective features are difficult to extract and encode. On the other hand, content-based features have been used for some time in retrieval and recommendation systems [4, 13], presenting some advantages: they can be automatically extracted and used to compute the similarity between songs; they help solving the problem of cold-start for new songs. Therefore, they constitute a good approach to characterizing activities and empower a recommendation solution.

To associate songs with activities and thus characterize them by using content-based features, we took a user-centered approach. To that end, we developed a web-application to collect songs that users enjoy listening to, while performing each activity (see Figure 1-1). The users selected songs first by filtering genres, then artists, and finally by songs (see Figure 2). Regarding genres, we adopted the taxonomy used by the majority of digital music services (like Musicovery[2]), showing only 15 different genres (Rock, Electro, Pop, R&B, Rap, Metal, Classic, etc.). After selecting one or more genres suitable for each activity, users could choose from 30 artists in maximum (twice the number of genres). Next, users could finally choose songs from the artists previously selected (a maximum of 100 songs were shown). Top artists and songs were used for the selection, gathered through the online service EchoNest. The result of this process was an association between activities and a set of songs suitable to be listened by different users.

To collect this data we sent emails to contacts and spread the link for the application through social networks, namely, *Facebook* and *Google+*, to reach as many users as possible. 98 subjects used the application, providing a total of 251 answers for all the activities. Despite the fact that some users did not provide feedback about their tastes for all activities, the distribution was uniform: 55 answers for the activity *walk*, 53 for *running*, 47 for *sleeping*, 48 for *relaxing* and 48 for *shopping*. This resulted in associating *8,518* songs with the activities.

To characterize the activities we extracted content information from the songs selected by users using the *Echonest* service. For performance issues, we opted to use only the top-100 preferred songs for each activity.

After extracting all the features offered by EchoNest, we performed an evaluation to measure how discriminative each feature was in this characterization. This evaluation was performed using the *CfsSubsetEval* attribute evaluator along with the *best-fit* search method from Weka[3] [8]. The following four features were selected as the most discriminative: *accousticness*, *energy*, *loudness* and *tempo*. Therefore, for each song we created a 4-feature vector describing its content, and for each activity an array of feature vectors of the songs associated with them (see Figure 1-2).

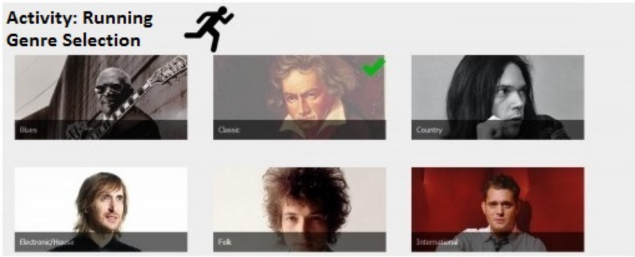
## 3.3 Generic Recommendation Model

The association detailed in the previous subsection allows us to describe each activity through a set of feature-vectors, representing each vector a song chosen by the users. To use this information in the suggestion of songs we need to convert it into a simpler representation to facilitate the rec-

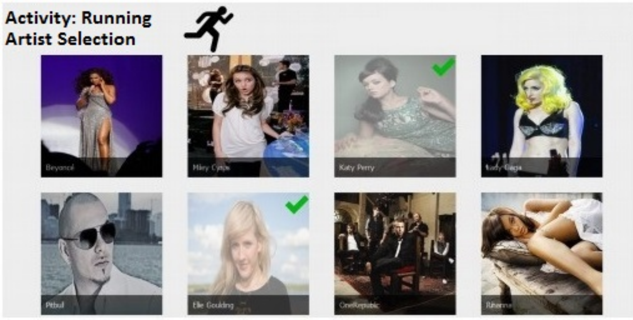
[1] http://the.echonest.com/

[2] http://musicovery.com

[3] http://www.cs.waikato.ac.nz/ml/weka/




(a) Genre selection.




(b) Artist selection.


(c) Songs selection.

Figure 2: Web-application developed to associate songs with activities.

ommendation process.

Typically, a recommendation system is often seen as a suggestion of items similar to a feature vector that represents the users' tastes. However, this approach is very restrict, since it will tend to recommend the same set of songs every time. Based on this and on the mechanism we are using to characterize each activity, we decided to use an interval approach for the recommendation.

To this end, we defined a set of intervals delimiting the value that each feature could take, instead of considering a single point in space (computed using a clustering algorithm, for instance). Moreover, this approach not only increases the range of songs that we can suggest for each activity, but also gives the possibility of using different sub-intervals to restrict the filtering process. This set of intervals define what we labeled as the *hyper-rectangle* (see Figure 1-3). A hyper-rectangle has four dimensions, and is defined by intervals with a maximum and a minimum value that each feature can take within each activity. The size and position of these rectangles differ between activities and for each user, pro-

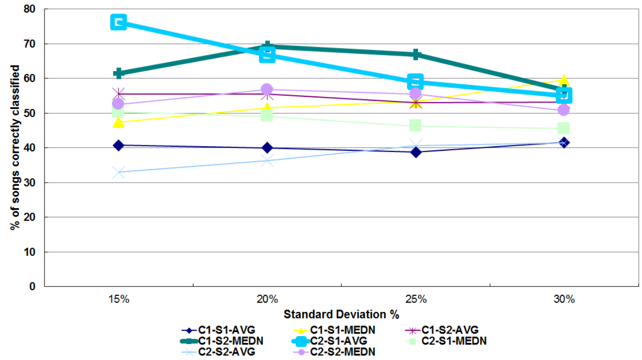



Figure 3: Method for determining the hyper-rectangle limits.

viding an adaptable recommendation mechanism as detailed later in the paper. The hyper-rectangles represent the backbone of Improvise. To recommend songs using them, we search for songs within the limits of the hyper-rectangles, using the *EchoNest* service.

Our generic recommendation approach consists in creating five generic hyper-rectangles, one for each activity, based on the songs collected through the web-application developed (see Section 3.2). This generic model is therefore capable of suggesting songs for each activity to any user, without having previous knowledge about him/her. This way we have a simple approach that provide an answer to the *cold-start* problem.

To calculate the intervals for each feature and thus define the hyper-rectangles, we started by testing two different methods. The first method (*M1*) used the average minus the standard deviation for finding the minimum of the interval and the average plus the standard deviation to find its maximum. The second method (*M2*) used the 10% percentile as the minimum value of the interval and the 90% percentile for its maximum. To evaluate the quality of the two methods, we searched for songs within the intervals defined, using the minimum and maximum values of the intervals for each feature. The results of these tests lead us to conclude that the methods were not adequate since the intervals generated were too wide, with a considerable overlap between them, blurring the differences between the recommendations for the different activities.

Therefore, we created two new methods: the first based on *M1* using a percentage of the standard deviation, with values of 15, 20, 25 and 30%; and the other method, similar to the previous one, but using the median instead of the average (*M4*). The limits for the hyper-rectangles were generated in two different ways: one using the top-100 songs selected by users, and the other using only the top-20. These variants, generated a set of 8 different data sets that were used to assess the quality of the proposed methods for the hyper-rectangle calculus. The datasets were used for training a Random Forest classifier with the goal of evaluating the quality of the limits generated (the adequacy of the songs to the activity). Figure 3 depicts the accuracy values of the classifiers. In this figure, C1 and C2 encode the datasets used to train the Random Forest classifiers: C1 represents the top-20 songs dataset, and C2 the top-100. Notice that C1 is a subset of C2, as these songs were those selected by users. The labels S1 and S2 encode the datasets used for determining the intervals of the hyper-rectangles: S1 indicates that the top-20 songs were used, while S2 represents the usage of the top-100 songs dataset. Again, S1 is a subset of S2. Finally, AVG stands for the average, while MEDN for the median.

We used these two dataset divisions (S1 and S2) to understand if it would be beneficial to have a wider (100) or narrower (20) history of songs for generating the intervals. Although the best result was achieved with the average $\pm 15\%$ of the standard deviation, the number of songs suggested by this method was smaller while compared to others. Therefore, we chose the second best method, which corresponds to the usage of the median $\pm 20\%$ of the standard deviation. In this case the median and standard deviation were calculated using the top-100 songs chosen by the users. The training instances used by the classifier were the top 20 songs chosen by the users for each activity (*C1-S2-MEDN*).

In summary, to create the generic recommendation model we defined a set of five hyper-rectangles, one for each activity, using the top-100 songs and the median $\pm$ 20% of the standard deviation as the method to determine their intervals. Thus, without previous knowledge about a user's preferences, we can generate recommendations suitable for him/her and for the activity at hand (see Figure 1-4).

### 3.4 Personalized Recommendation Model

To personalize the recommendations for each activity we incorporate the user feedback, expressed by selecting the songs she/he considered adequate for the activity. This is then materialized by adjusting the intervals for each activity based on the songs listened.

While the method for determining the hyper-rectangles in the personalized model is the same as in the generic approach (top-100 songs and the median $\pm$ 20% of the standard deviation), the list of songs used is different. This list starts with the top-100 songs chosen by all users (and used to create the generic model) and is updated with the new songs selected by the users. These are added to the end of the list replacing the oldest ones, as they represent less preferred songs.

When the list of songs used to generate the intervals no longer contains songs used for the generic model, the process follows a FIFO order (*First In First Out*). This approach constantly personalizes the recommendation model by considering the user feedback and by adjusting to his/her current tastes and preferences, over time. New songs remain more time in the list used to determine the new intervals. This design allows us to perform a more personalized recommendation, taking advantage of the current tastes and preferences of the users.

## 4. EVALUATION

We conducted two user-centric experiments to evaluate both recommendation approaches offered by Improvise, the generic and the personalized model. To that end, we developed a web application where users could select the songs they consider appropriate for each activity. By counting the number of suitable songs we could measure the effectiveness of Improvise in suggesting songs for daily activities.

In the following sections we describe our objectives, the participants, the evaluation procedure, the main results and the discussion about them.

### 4.1 Goals and Tasks

The main goal of Improvise is to recommend and suggest songs to be listened while doing activities, such as, running, relaxing or shopping. To validate both the generic and the personalized solution, we divided the evaluation into two phases.

The first phase consisted in evaluating the generic model to understand if it was flexible enough to recommend music that fit the preferences of any potential user. In the second phase, the songs selected by each user during the first evaluation (feedback) were used to individually personalize Improvise and to generate new recommendations for each activity. The main objective consisted in understanding if personalized suggestions were better than those generated using the generic model. Finally, a second interaction with the personalized model was conducted to assess the impact in personalization over time. Here, the personalized recommendation model suffered a second personalization by taking into account the new feedback collected during the previous session.

The main task for both phases consisted in selecting the appropriate songs for each activity from a list of songs suggested by our solution.

### 4.2 Participants

During the first phase of the evaluation, ten users participated in the experiment. Eighty percent of the subjects were male, with ages between 22 and 29 years old (90%), being graduate or undergraduate students from the university campus. All of them reported listening to music for different activities during the day.

In the first iteration of the second experiment all the ten previous subjects participated in the tests using their personalized version of the hyper-rectangles for each activity, created based on their feedback from the first phase. Due to time restrictions, only five of the ten users were able to participate in the second iteration of the second phase. Here, we used a new personalized version of the recommendation model, created using the feedback provided in the previous session.

### 4.3 Procedure

To evaluate the proposed solution we developed a web application for users to interact with the recommendation technique (see Figure 4). For both experiments, the evaluation started first with users answering a small questionnaire with demographic information to characterize them (*e.g.* age, gender, music listing information, etc.). Then users selected the appropriate songs for each activity, and at the end they filled a satisfaction questionnaire. Notice that the activities


(a) Activity selection. (b) Song selection.

Figure 4: Application developed for evaluating the recommendation model.

were simulated, as the users were not actually performing them, we just mentioned their names.

To evaluate the adequacy of the generic and personalized models, we presented 50 songs for each activity, from which users should select those they consider correctly assigned to the activity. Songs were presented (album cover, song and artist name) one at a time, with the possibility of playing a 30 seconds sample.

After selecting the songs for each activity, users were asked to answer a satisfaction questionnaire to express their agreement with the suitability of the suggested songs for the activity in question.

## 4.4 Results

Overall, users were satisfied with the recommendations performed by both the generic and the personalized model. Moreover, the effectiveness of the personalized model was confirmed by a steady increase in the number of songs considered suitable for each activity by the users, from the generic model to the personalized model.

On average users selected eleven to twelve songs, for each activity, from the list suggested by the generic model, corresponding to 24% of the total of songs recommended.

For the first iteration of the personalized model, as depicted in Figure 5, users selected on average more than 15 songs

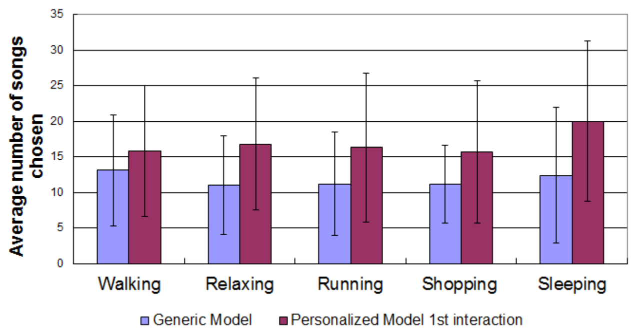



Figure 5: Comparison between the generic and the personalized model in terms of the number of songs selected for each activity. Error bars denote standard deviation.

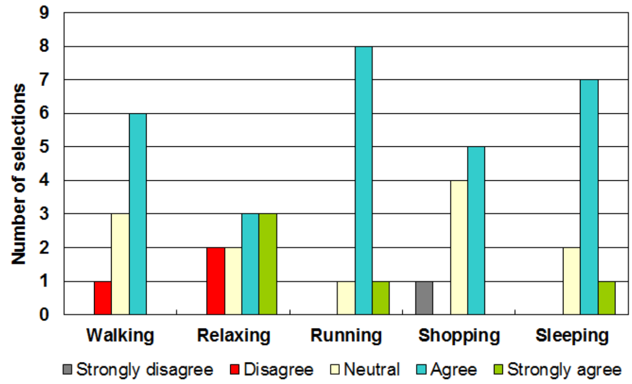



Figure 6: Results of the user satisfaction regarding songs suggested by the generic model.

per activity (30%). This corresponds to an average increase of 25% over the number of songs selected using the generic model. *Sleeping* is the activity that presents the best results and the highest improvement for the personalized model.

The satisfaction questionnaire used to collect users opinion about the quality of the suggested songs was composed by a five point likert scale, with answers as *strongly disagree, disagree, neutral, agree and strongly agree*. Figure 6 depicts the results for the generic model. Overall, more than half of the users agreed or strongly agreed with the suggested songs, for four of the five activities. Only the *Shopping* activity did not achieve this value.

Figure 7 depicts the total number of songs considered appropriate for the various activities. As we can see, there is a steady increase in the number of correct songs, from the generic model to the second iteration of the personalized model. Indeed, this corresponds to an increase of 31% (on average) for all users, revealing that our model is able to fit the tastes of the different users over time.

The growth in the number of songs from the first to the second iteration of the personalized model was around 10%. Detailed data on the behavior of the three models, for the five users who participated in the three test sessions, is depicted in Figure 8. As we can see, overall, the personalized

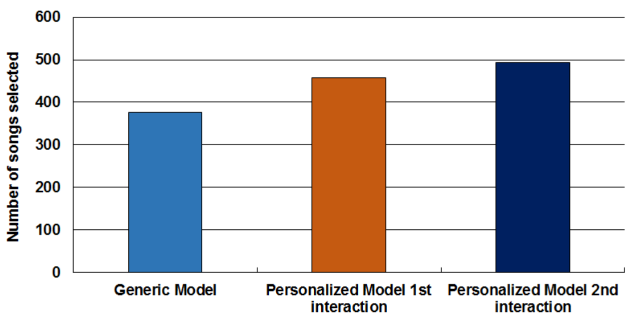



Figure 7: Evolution of the total number of songs selected for the various activities using the different recommendations models.

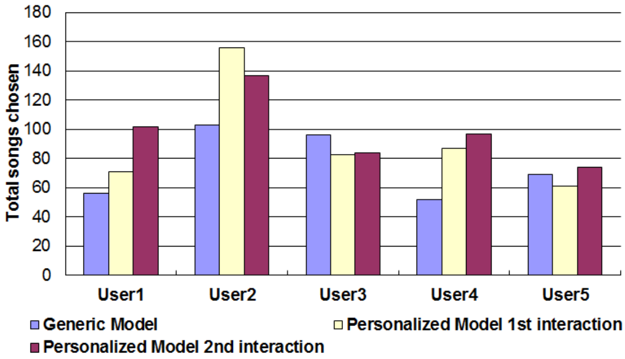



Figure 8: Results of the evaluation of the personalized model for each user.

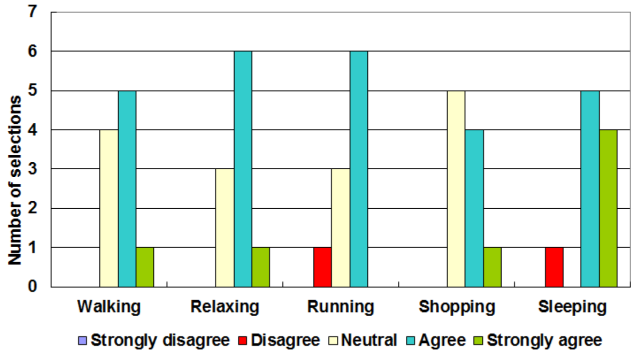



Figure 9: Results of the user satisfaction regarding songs suggested by the personalized model during the first iteration.

models suggested more songs suitable for the different activities than the generic model (Users 1, 2, 4 and 5). Only for User 3 the personalized model suggested less adequate songs. Two other special cases are worth mentioning: for User 2 the second iteration with the personalized model performed worse than in the first iteration; and for User 5, the personalized model required a second iteration to outperform the generic model.

Figure 9 depicts the satisfaction results for the first iteration of the personalized model. Similarly to what happened with the generic model, more than half of the users agreed or strongly agreed with the suggested songs, for four of the five activities. But, for the personalized model we have more strongly agree answers. The *Shopping* activity still has the worst results, but are better than in the generic model.

To get a better understanding of the improvement provided by the personalized model, we grouped the users' answers about the generic and the personalized model in negative (strongly disagree and disagree), neutral and positive (agree and strongly agree) opinions. We found an increase of 13% in the number of positive opinions from the generic to the personalized model, showing that the personalized suggestions are more inline with users' preferences.

### 4.5 Discussion

From these preliminary results, we can conclude that our work is on the good path to create an approach able to effectively suggest songs for daily activities and flexible enough to adapt the recommendation list to the users' tastes and preferences over time, supporting both "unknown" and "known" users.

Results for the number of songs chosen for each activity show that users selected more songs while using the personalized model than while using the generic model. This confirms that our solution can effectively suggest songs for different users and activities, and adapt to their preferences. Moreover, the second iteration with the personalized model reinforced these results. Satisfaction results were also in agreement with the reported increase in the number of songs selected. Users were overall happy and satisfied with the recommendations performed.

In a particular case the personalized model required two iterations to get adjusted to the user, showing that for some users our model needs more time to "learn" the users preferences. In another case, for which we did not find any evidence for it, the user preferred more songs from the generic model than from the personalized ones.

Although these results are very promising, showing that our approach can deal with the cold-start problem by providing a generic model that can suggest songs for any user without knowing anything about them, we would like to mention some constraints that prevent us from state stronger claims. First, we cannot draw any statistical significance from the results due to the small number of users involved in the preliminary evaluation. In a near future we plan to perform an evaluation with a larger number of users. Second, users were not performing the activities for which we suggested songs. More evaluation is required to clarify if this affected the result.

## 5. CONCLUSIONS AND FUTURE WORK

Nowadays, a huge amount of songs is available to millions of users around the world. With millions of artists and songs on the market, it is difficult for users to find songs that please them. This problem is even worse when trying to select songs for different activities.

In this paper we described Improvise, an adaptable solution for recommending songs for daily activities. Improvise is a user-centered approach that relies on the hyper-rectangle concept, determined using content from songs. We described the rationale behind the calculus of the hyper-rectangles for a generic recommendation model and also the creation of a personalized solution. Preliminary results show that the generic model was successful in recommending songs to users. But more relevant is the flexibility of the solution in adapting the recommendation to different users for each activity, increasing not only the number of songs selected, but also their satisfaction.

Regarding future work, we plan to explore two paths. The first is to explore new and different methods for determining the hyper-rectangles, like for instance to consider more than one hyper-rectangle for each activity. This can capture more diverse and sparse preferences and tastes, promoting new recommendations and user satisfaction. The second path is to use a larger number of songs to determine the hyper-rectangles that characterize the user profile, since at the moment we are only using the top-100 songs preferred by the users as detailed in Section 3. Although, using more songs could bring a more accurate and detailed calculus of the hyper-rectangles, a study to determine the best number of songs is also planned.

In summary, we can report that Improvise suggests songs suitable for daily activities to users with different tastes and preferences. Moreover, the proposed model is flexible enough to constantly adapt its recommendations accordingly to the user feedback, while providing an answer to the cold-start problem.

## 6. ACKNOWLEDGMENTS

This work was supported by national funds through Fundação para a Ciência e Tecnologia, under INESC-ID multi-annual funding - PEst-OE/EEI/LA0021/2013 and LaSIGE Strategic Project - PEst-OE/EEI/UI0408/2014. Ricardo Dias was supported by FCT, grant reference SFRH/BD/70939/2010.

## 7. REFERENCES

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Proc. of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999.

[2] L. Baltrunas, X. Amatriain, and V. Augusta. Towards Time-Dependant Recommendation based on Implicit Feedback. In *Context Aware Recommendation Systems (CARS - RecSys)*, 2009.

[3] R. Cai, C. Zhang, C. Wang, L. Zhang, and W.-Y. Ma. Musicsense: contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 553–556, New York, NY, USA, 2007. ACM.

[4] P. Cano, M. Koppenberger, and N. Wack. An industrial-strength content-based music recommendation system. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in Information Retrieval*, 2005.

[5] O. Celma. *Music Recommendation and Discovery.* Springer, 1st edition, 2010.

[6] W. W. Cohen and W. Fan. Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks*, 33, 1999.

[7] R. Dias and M. J. Fonseca. Improving music recommendation in session-based collaborative filtering by using temporal context. In *ICTAI*, 2013.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 2009.

[9] B. P. Knijnenburg, S. Bostandjiev, J. O'Donovan, and A. Kobsa. Inspectability and control in social recommenders. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, 2012.

[10] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, J. Riedl, and H. Volume. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 1997.

[11] T. Leong, S. Howard, and V. Frank. Choice: Abdicating or exercising. In *CHI*, 2008.

[12] N. Liu, S. Lai, and C. Chen. Adaptive music recommendation based on user behavior in time slot. *Journal of Computer Science and Network Security*, 9(2):219–227, 2009.

[13] D. McEnnis and S. J. Cunningham. Sociology and music recommendation systems. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 185–186, 2007.

[14] S. E. Park, S. Lee, and S.-g. Lee. Session-Based Collaborative Filtering for Predicting the Next Song. *First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, 2011.

[15] S. Reddy and J. Mascia. Lifetrak: Music in tune with your life. In *Proceedings of the 1st ACM International Workshop on Human-centered Multimedia*, 2006.

[16] M. Schedl and P. Knees. Personalization in Multimodal Music Retrieval. In *9th International Workshop on Adaptive Multimedia Retrieval*, 2011.

[17] B. Schwartz. *The Paradox of Choice: Why More Is Less.* Harper Perennial, 2005.

[18] B. Shao, M. Ogihara, D. Wang, and T. Li. Music recommendation based on acoustic features and user access patterns. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(8):1602 –1611, nov. 2009.

[19] J. Su, H. Yeh, and P. Yu. Music recommendation using content and context information mining. *Intelligent Systems, IEEE*, (IEEE Intelligent Systems), 2010.

[20] K. Verbert, D. Parra, P. Brusilovsky, and E. Duval. Visualizing recommendations to support exploration, transparency and controllability. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, 2013.

[21] X. Wang, D. Rosenblum, and Y. Wang. Context-aware mobile music recommendation for daily activities. In *Proc. of the 20th international conference on Multimedia*, 2012.

[22] B. Yapriady and A. L. Uitdenbogerd. Combining demographic data with collaborative filtering for automatic music recommendation. Springer, 2005.

[23] E. Zheleva, J. Guiver, E. Mendes Rodrigues, and N. Milić-Frayling. Statistical models of music-listening sessions in social media. *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010.