A MULTIOBJECTIVE MUSIC RECOMMENDATION APPROACH FOR ASPECT-BASED DIVERSIFICATION

Ricardo S. Oliveira **Caio Nóbrega** **Leandro B. Marinho** **Nazareno Andrade**

UFCG - Federal University of Campina Grande, Brazil

{ricardooliveira, caionobrega}@copin.ufcg.edu.br, {lbmarinho, nazareno}@computacao.ufcg.edu.br

ABSTRACT

Many successful recommendation approaches are based on the optimization of some explicit utility function defined in terms of the misfit between the predicted and the actual items of the user. Although effective, this approach may lead to recommendations that are relevant but obvious and uninteresting. Many approaches investigate this problem by trying to avoid recommendation lists in which items are very similar to each other (aka diversification) with respect to some aspect of the item. However, users may have very different preferences concerning what aspects should be diversified and what should match their past/current preferences. In this paper we take this into consideration by proposing a solution based on multiobjective optimization for generating recommendation lists featuring the optimal balance between the aspects that should be held fixed (maximize similarity with users actual items) and the ones that should be diversified (minimize similarity with other items in the recommendation list). We evaluate our proposed approach on real data from Last.fm and demonstrate its effectiveness in contrast to state-of-the-art approaches.

1. INTRODUCTION

In scenarios of vast and dynamic availability of content, such as online music streaming services, users are quickly overloaded with a large and ever increasing space of choices. Recommender systems are successful tools for addressing this issue by modeling the preferences of users and anticipating their information needs. The most successful recommendation approaches are usually those based on the optimization of some explicit utility function defined in terms of the misfit between the predicted and the actual items consumed by the user. Although effective in many scenarios, the recommendation algorithms that optimize this kind of function are prone to deliver recommendations that are relevant but possibly uninteresting. For example, for a user who only listens to American punk rock bands from the 70's, a recommendation of more bands of this kind would probably be accurate, but possibly tedious given that this user may very likely be able to find these artists without aid.

In order to mitigate this problem, many approaches have appeared with the aim of increasing diversity in recommendations [6, 14, 17, 19]. This is usually achieved by mechanisms that avoid recommendation lists in which items are very similar to each other with respect to some aspect of the items (e.g. music genre). Such approaches can potentially increase users satisfaction by providing less obvious recommendations. However, users may have different preferences concerning what aspects should be diversified and what should match their past/current preferences. For example, a user may be very conservative concerning the music genres she likes to listen (e.g. Bossa Nova), but very open to discover how this genre is played across different countries (e.g. Bossa Nova played in Japan and India). This is exactly the recommendation scenario we investigate in this paper, i.e., we want to generate recommendation lists by explicitly holding one or more item aspects (e.g. Bossa Nova) constant, but increasing diversity in others (e.g. locality and time period). The aspects that are held fixed are the ones that correspond to the users past/current preferences while the others correspond to the way the users are open for diversification.

This problem has two different and possibly conflicting objectives that need to be optimized for each user's recommendation list: (i) find the items that maximize the similarity with the preferences of the user in terms of some set of selected aspects, and (ii) find the items that minimize the intra-list similarity (i.e. pairwise similarity of items in the recommendation list) regarding a different set of selected aspects. To generate recommendation lists that balance these two objectives we propose Multiobjective Aspect Diversification (MOAD), a recommendation approach that cast this problem as a multiobjective optimization problem. MOAD uses the Nondominated Sorting Genetic Algorithm - NSGA-II, which is an efficient solver for this kind of problem [5]. The main difference between our approach and other related work from the literature is that we allow the explicit choice of the aspects to diversify and hold constant.

Although music recommendation may refer to many distinct entities in the literature, such as song tracks, albums and artists, in this paper we focus on artists due to their relative abundance of aspects publicly available. In order to assess the effectiveness of our proposed approach, we collected artist listening historical data from Last.fm [1], a large online radio portal, and enriched this collection with artists metadata collected from Music Brainz [2] and DBPedia [3]. We conduct several experiments by considering several switches between the fixed and variable aspects and show that MOAD achieves the sought balance between both objectives. We also compare our approach with several diversification algorithms from the literature and show that our recommendations are more diverse and relevant in the chosen aspects in comparison to those.

2. PROBLEM FORMALIZATION

The problem that we address in this paper can be stated as follows: given a target user $u \in U$ (where U is the set of users), her item consumption history $I_u \subseteq I$ (where I is the set of items) [4], two disjoint sets X and Y of item aspects (possibly provided by the user), we want to find the top-n items that are more similar to I_u regarding X and more dissimilar from each other regarding Y.

Items may have different kinds of metadata associated to them, aka attributes, dimensions, contexts or side information. For example, if the recommendable item is a music artist, we can think of each possible music genre as a binary attribute. More formally, let G be the set of genres and $d : G \times I \to \{0, 1\}$ a function that indicates if genre $g \in G$ describes item $i \in I$. The set G is thus one aspect of the item and represents the set of attributes of the kind genre.

More generally, let $A = \{A_1, A_2 \ldots, A_t\}$ denote the set of t possible aspects and $f_j : A_j \times I \to \mathbb{R}$ be a feature extractor for any attribute $a \in A_j$ that describes item i. An item may now be represented as a feature vector regarding aspect j. For example, $\vec{i} = (f_1(a_1, i), f_1(a_2, i), \ldots, f_1(a_p, i))$ represents a feature vector of item i where $a_1, a_2, \ldots, a_p \in A_1$.

The input for the algorithm is a user $u \in U$, her consumption history I_u, the size n of the recommendation list and two disjoint sets of aspects: $X, Y \subseteq A$ where $X \cap Y = \emptyset$. We also consider two similarity functions: $g_a(R)$ that returns the intra-list similarity (cf. Section 4.4) of the items in the recommendation set R w.r.t aspect a; and $h_a(R, I_u)$ that returns the similarity between R and the user history I_u. Finally, let

$$\textit{diversity}(R, X) = 1 - \frac{1}{|X|} \sum_{a \in X} g_a(R) \tag{1}$$

denote the average intra-list distance for all aspects in X and

$$\textit{affinity}(R, Y) = \frac{1}{|Y|} \sum_{a \in Y} h_a(R, I_u) \tag{2}$$

the average similarity between R_u and I_u for all aspects in Y. Now, given the aforementioned inputs, we want to find a set $R \setminus I_u \subseteq I$ of n items (i.e. $|R| = n$) that maximize, at the same time, the objective functions defined in equations 1 and 2, i.e.:

$$\arg\max_R \left(\textit{diversity}(R, X), \textit{affinity}(R, Y)\right) \tag{3}$$

3. RELATED WORK

Several works have appeared in recent years proposing recommender systems concerned with other metrics beyond accuracy such as diversity, novelty or serendipity. Most of these works aim to maximize such an alternative metric without degrading accuracy. The seminal work of Ziegler et al. [19] laid the foundations for achieving diversity based on a re-ranking of collaborative filtering algorithms results. Several other works appeared following similar principles but based on different techniques such as graphs [8], machine learning [7, 18] and information retrieval [15].

Another strand of work considers this problem as a multiobjective optimization task. Realizing that accuracy, diversity and novelty might be conflicting objectives, Ribeiro et al. [11] proposed a hybrid recommendation system that combines algorithms through an evolutionary approach to maximize one objective, without sacrificing the others. Ouni et al. [9] proposed a genetic algorithm to recommend software libraries, finding a trade-off between three objectives. Wang et al. [16] developed a multiobjective solution to recommend accurate and unpopular items, called long tail recommendations. Zuo et al. [20] proposed personalized recommendations by balancing accuracy and diversification. And finally, Pampalk and Goto [10] proposed a graphic interface where users may adjust the recommendations received according to her desire by adjusting music aspects. Our work also use multiobjective evolutionary algorithms for promoting diversification, but differently from the aforementioned related works, we enable the explicit specification of the aspects that should be diversified.

4. MULTIOBJETIVE ASPECT DIVERSIFICATION

The main motivation for this research is to give users more control on their recommendations. We do this by letting the aspects that should be held constant and the ones that should be diversified user-definable. In this section we describe in detail the components of our approach.

4.1 Pareto Optimality

In multiobjective optimization the best solutions are the ones that cannot be improved in any of the objectives without degrading at least one of the other objectives. This property is known as *Pareto optimality*. In our case, a feasible solution $R \subseteq I$ (i.e. a recommendation list of size n) is said to dominate another solution $R' \subseteq I$ if:

1. $\textit{div}(R, X) \geq \textit{div}(R', X) \wedge \textit{aff}(R, Y) \geq \textit{aff}(R', Y)$

[1] http://www.last.fm/
[2] https://musicbrainz.org/
[3] http://wiki.dbpedia.org/
[4] I_u can also be thought as a query in terms of MIR

2. $div(R, X) > div(R', X) \vee aff(R, Y) > aff(R', Y)$

where *div* and *aff* are abbreviations for *diversity* and *affinity* respectively. A solution $R^* \subseteq I$ is called Pareto optimal if there is no other solution that dominates it. The set of Pareto optimal solution is also known as the Pareto front.

4.2 Evolutionary Algorithm Approach

Determining the Pareto front is known to be a computationally intensive task [2, 11]. In our case it basically requires an enumeration of all possible recommendation lists for the objectives evaluation (i.e. $\mathcal{O}(2^{|I|})$). Among the approaches used for addressing this kind of problem, evolutionary algorithms appear as the most efficient and used ones [4, 12, 16, 20]. Thus, we have decided to adopt a Multiobjective Evolutionary Algorithm (MOEAs) for addressing our research problem.

The idea is to generate a population of recommendation lists as individuals such that the dominating individuals are considered the fittest and are kept for the next generation. If the dominating individuals are insufficient to compose the new generation, some dominated individuals are chosen to compose the next generation. The crossing over - switching the items between neighbor individuals - and the mutation probability - used to replace some random items in individuals for items still not considered - allow new individuals to approach the Pareto front throughout several generations.

4.3 NSGA-II

Similarly to other related work, we have chosen the Non-dominated Sorting Genetic Algorithm II (NSGA-II) as the MOEA solution [9, 20]. Besides giving guarantees of convergence, it also offers a fast sorting function, called Fast Non-dominated Sorting, with $\mathcal{O}(MN^2)$ where M is the number of objectives and N is the population size. This sorting function separates individuals into levels of dominance. For individuals in the same level, NSGA-II estimates the density of solutions, privileging a set of solutions that are spread on the objective space, in a process called Crowding Distance Assignment.

Algorithm 1 presents NSGA-II pseudocode. The algorithm starts by creating an initial population of size N (line 2). The following steps are repeated for each generation. A new offspring is created based on the current population (line 4) and the individuals are ordered and selected to compose the population for the next generation (lines 5 to 11). This ordering considers first the selection of individuals whose objectives are not dominated by other individuals, made by *fast-nondominated-sort* (line 6), and second, the density of individuals provided by *crowding-distance-assignment* (line 8). If the non-dominated individuals are not enough to complete N, then individuals on the second level of dominance are chosen, and so on (lines 12 and 13). The population P_{t+1} is the output for the algorithm, and we select the individual with the greater sum of objective values as the final recommendation list.

Algorithm 1 NSGA-II

```
1:  procedure NSGA-II(N, nGen, mProb, cProb)
2:      P_0 = create-initial-population[N]
3:      for t in 0 to nGen -1 do
4:          Q_t = create-new-offspring(P_t, mProb, cProb)
5:          R_t = P_t + Q_t
6:          F = fast-nondominated-sort(R_t)
7:          while |P_{t+1}| + |F_i| ⩽ N do
8:              crowding-distance-assignment(F_i)
9:              P_{t+1} = P_{t+1} ∪ F_i
10:             i = i + 1
11:         end while
12:         Sort(F_i, ≺)
13:         P_{t+1} = P_{t+1} ∪ F_i[N − |P_{t+1}|]
14:     end for
15: end procedure
```

4.4 Item Representation and Similarity Metrics

In order to compute the objective functions defined in equations 1 and 2 we need to compute similarities between items concerning the sets of aspects used as input. Thus, we define feature extraction functions for each aspect such that similarity measures can be applied.

4.4.1 Aspects Definition

First we need to instantiate the set A of aspects that we consider in this paper:

- **Contemporaneity (A_1):** refers to the year the artist was born (if the artist is solo) or the year the band was formed, in case the artist is a band.
- **Locality (A_2):** refers often, but not always, to artist's birth/formation country.
- **Gender (A_3):** refers to the artist gender (when applicable) together with its type (i.e. solo, band, orchestra, etc.). This aspect is a combination of two aspects where if the artist type is *person* (i.e. a solo artist) its gender is *male* or *female*. Otherwise, it has no gender but has a type that can be one of the following: *group*, *orchestra*, *choir*, *character* or *other*.
- **Music Genre (A_4):** refers to the artist music genres.

We have chosen this aspects for two main reasons: (i) they are used recurrently in related works (not necessarily together) as side information for improving the preference modeling of users; and (ii) they are publicly available in MusicBraiz and DBpedia.

4.4.2 Similarity Metrics

Regarding A_1, each item is represented as one-dimensional vectors in which their single component is the year normalized to the range $[0, 1]$. More formally, for a given contemporaneity (i.e. year) $a \in A_1$ associated to artist i

$$f_1(a, i) = \frac{a - \min(A_1)}{\max(A_1) - \min(A_1)}$$

where $\min(A_1)$ and $\max(A_1)$ returns the minimum and maximum contemporaneity values of A_1 respectively. Now, the similarity of two items i and j regarding their respective contemporaneities $a_i, a_j \in A_1$ is simply computed as the inverse of their distance, i.e.,

$$sim_{A_1}(i,j) = 1 - (\vec{i} - \vec{j}) \tag{4}$$

where $\vec{i} = (f_1(a_i, i))$ and $\vec{j}$ is defined analogously. The intuition here is that artists from the same time epoch tend to produce similar music.

Concerning A_2, the feature extraction f_2 is basically an identity function, i.e., a function that returns the same value found in the raw data. So, similarly to A_1, items are represented as one-dimensional vectors whose single component is a nominal value (e.g. country name). For computing similarities between items under this representation we used the Occurrence Frequency (OF) metric [1] which besides being suitable for categorical data, exploits the frequency of items with regard to the associated features. This metric assigns 1 to items having the same feature value and different scores to mismatches. A mismatch between less frequent items regarding their features yields a lower value than a mismatch between more frequent items. For example, if we compare two artists from USA and England respectively, two countries with a large number of artists in the dataset, their similarity will be greater than artists of USA and Costa Rica, since Costa Rica has probably much less artists than USA. The idea is basically to avoid having zero similarities whenever a mismatch occurs.

The equation below defines OF (which is used as sim_{A_2}) of two items i, j regarding *Locality*:

$$OF_{A_2}(i,j) = \begin{cases} 1 & \text{if } \vec{i} = \vec{j} \\ \frac{1}{1 + log\ \frac{|I|}{freq_{A_2}(i)} \times log\ \frac{|I|}{freq_{A_2}(j))}} & \text{otherwise.} \end{cases} \tag{5}$$

where $freq_{A_2}(i)$ returns the number of artists having the same feature value (country name in this case) as item i.

Regarding A_3, f_3 is analogous to f_2, but the item representation is slightly different. Since *Gender* is actually a combination of two aspects, items are represented by vectors containing two nominal values: (*type*,*gender*) where gender can be *male* or female if the artist type is *person*, and *neither* if the artist is associated to any other type. For calculating the similarity between items i, j we apply equation 5 separately for *type* and *gender* and take the average. More formally,

$$sim_{A_3}(i,j) = \frac{OF_{type}(i,j) + OF_{gender}(i,j)}{2} \tag{6}$$

As an example, the similarity between a male singer and a female singer should return a greater similarity than between a male singer and a band.

As an artist can be associated to multiple music genres, the feature extraction function for A_4 is the function $f_4 : A_4 \times I \to \{0, 1\}$ that indicates the genres that are associated to a given artist. Each item is then represented by a binary vector of genres. To measure similarity between two items i, j regarding this aspect we use the well known cosine similarity function, i.e.,

$$sim_{A_4}(i,j) = cos(\vec{i}, \vec{j}). \tag{7}$$

Now, we can finally instantiate functions $g_a(\cdot)$ and $h_a(\cdot)$ introduced in section 2. For a given aspect $a \in A$, a recommendation list R and I_u:

$$g_a(R) = \sum_{(i,j) \in R \times R | i \neq j} sim_a(i,j) \tag{8}$$

$$h_a(R, I_u) = \sum_{(i,j) \in R \times I_u | i \neq j} sim_a(i,j) \tag{9}$$

5. EVALUATION

In this section we evaluate the effectiveness of MOAD for music recommendation. All code for the evaluation is available publicly online [5] .

5.1 Data Collection and Preparation

We used three publicly available data sources: Last.fm, Music Brainz and DBpedia. Last.fm is a social network where users share data about their listening habits. In particular, we have used a recent Last.fm dataset published and made available by Schedl [13].

For extracting the aspects about the artists available in the Last.fm dataset, we have used Music Brainz, a music encyclopedia that provides rich metadata about artists and albums. From Music Brainz we extracted *Contemporaneity*, *Gender*, and *Locality*. Finally, we used DBpedia to extract the *Music Genre(s)* of each artist.

After enriching the artists with the aforementioned aspects, genres associated with less than 5 artists were removed, as well as artists with no genre at all. Finally, a sample of 1,000 users from the Last.fm dataset was randomly selected for the experiments. This number of users is in line with the size of other very well known and used Last.fm datasets in the music recommendation community, see for example the *Last-fm - 1K users dataset* [6] [3]. In our sample, 3 users had no history and were thus discarded. We also generated a train/test time split where the first 80% of artists listened by each user was used for training and the remaining 20% for testing.

Our sample includes the following statistics: 14,415 artists, a median of 140 artists listened per user, 10 genders, 437 localities, 847 genres, and contemporaneity ranging from 1212 to 2014. Figure 1 shows a flow chart summarizing our approach.

5.2 Evaluation Protocol and Metrics

Ideally, the aspects to keep and the ones to be diversified should be provided by the users themselves. Since this kind of online experiment can be very demanding, it will be left for future work. In this paper, we will simulate some

[5] https://github.com/ricooliveira/moad.git

[6] http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html

possible scenarios and evaluate the extent to which MOAD can cope with them. The evaluation scenario is the following: one aspect is chosen to be diversified while the others are kept constant. Since we have four aspects, we end up with four evaluation scenarios. For example, a given user wants artist recommendations that are diverse regarding locality (artists countries) while maintaining genre, gender and contemporaneity constant (i.e. similar to previous listened artists w.r.t. these aspects).

Regarding evaluation metrics, we use *diversity* and *affinity* defined in equations 1 and 2 respectively. *Diversity* is actually related to a popular diversity metric known as ILD (Intra-List Diversity) [19] while *affinity* tries to assess the relevance of the recommendations regarding the aspects that were held constant. While *diversity* is only evaluated on the final recommendation list, *affinity* is evaluated on the test set.

#	Diversify	Maintain affinity
1	Cont.	Gender, Locality, Genre
2	Gender	Cont., Locality, Genre
3	Locality	Cont., Gender, Genre
4	Genre	Cont., Gender, Locality

Table 1. Evaluated recommendation scenarios

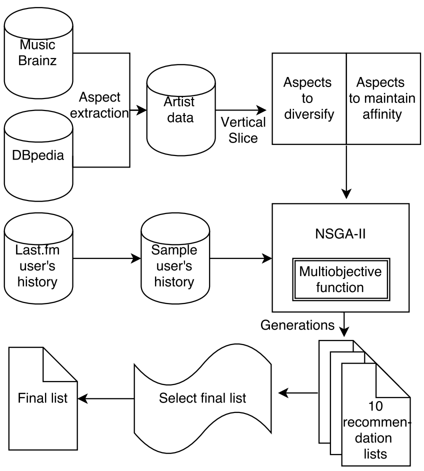

Figure 1. Flow chart of MOAD.

5.3 Baseline Algorithms

We have chosen baselines that are well known for promoting diversification without degrading accuracy. Since all of them compute item similarities in order to select the items that will compose the final recommendation list, we used the same similarity measures defined in section 4.4. This means that each baseline focused on the diversification of the same aspect, depending on the recommendation scenario chosen, as MOAD.

More specifically, we compare our approach to the following baselines:

- **Topic Diversification (TD):** Receives an initial recommendation list of 50 items where the first item of this list goes to the recommendation final list in order to preserve accuracy. Next, items are selected in an iterative and greedy fashion based on their rankings in the initial list and the similarity to the items already in the final list regarding the aspect of interest [19].
- **Relevance-based eXplicit Query Aspect Diversification (RxQUAD):** Performs a re-ranking over 50 precomputed items. During the greedy iterative step each item receives a score based on two factors: given an input aspect, the relevance of the aspect for the user and the relevance of the aspect for the item [15].
- **User-Based Collaborative Filtering (UBCF):** We also included a standard user-based collaborative filtering based on k-nearest neighbors. Notice that this algorithm is not aimed to promoting diversification.

We have used the RankSys tool where these three algorithm are implemented [14, 15].

5.4 Parameter Tuning

For determining suitable values to the NSGA-II parameters such as the number of generations, size of the population and probability of mutation, we extracted a subsample of 30 users from the Last.fm experimental dataset and determined a fixed scenario to perform some executions of our approach. We use the third scenario of Table 1 and $N = 10$, $nGen = 10$, $mProb = 0.1$ and $cProb = 0.9$ as default values. For tuning a particular parameter, we fixed the other parameters to its default values and varied the target parameter until no significant changes were found in the evaluation metrics. Due to the non-normality of the data, Wilcoxon non-parametric test was used. The tests determined that the ideal values to the parameters are $N = 10$, $nGen = 50$ and the default values to $mProb$ and $cProb$.

6. RESULTS AND DISCUSSION

For assessing MOAD variability across different executions, we run MOAD 10 times on scenario 3 of Table 1 and performed a Kruskal-Wallis test, which reported that there are no significant changes within the executions. We thus assume that other scenarios will follow a similar trend and thus only make one run of the algorithm for each subsequent scenario. The baseline algorithms are deterministic, so running them multiple times is not necessary.

For assessing the results we calculated *diversity* and *affinity* for all users in the experimental dataset. As mentioned earlier *affinity* was computed in the test set of each user. The boxplots of the results for all scenarios in Table 1 are shown in Figure 2. Notice that MOAD achieved better results than the baselines in all scenarios, considering both evaluation metrics, with a small variability across users. Wilcoxon tests are performed comparing MOAD to each baseline and all the differences are statistically significant albeit small in some cases.

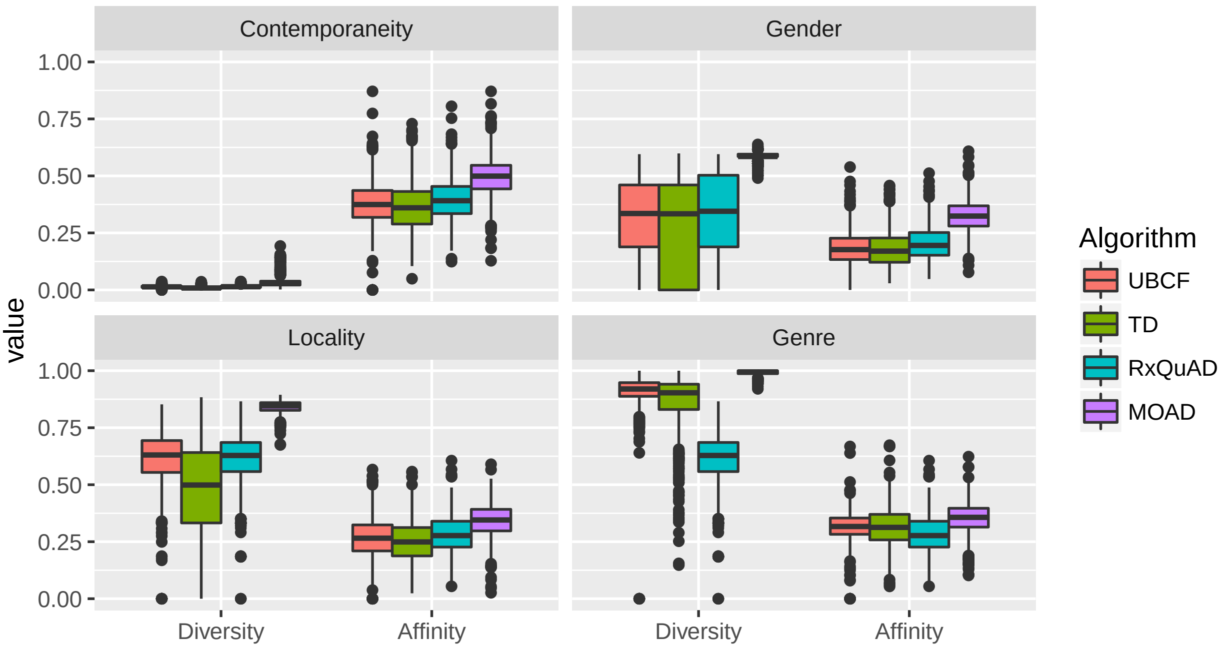

Figure 2. Comparison of MOAD and baselines

When diversifying *contemporaneity*, *diversity* shows very low results for all algorithms. This may be explained by the range of the aspect, mentioned on subsection 5.1, which turns the time difference, even between artists from different decades, very small when normalized. Gender is the scenario where the smallest differences between MOAD and the compared baselines are observed. A possible explanation to this is the fact that Gender aspect has only 10 possible values which does not leave much room for diversification. Locality and genre are the scenarios where we observed the highest gains, which is probably associated to the large number of possible values for these aspects.

Table 2 shows an example of a real Last.fm experimental user, receiving recommendations of three algorithms, based on scenario 3: UBCF; TD, the best baseline in scenario 3 and MOAD. In the simulations, MOAD obtained an improvement of 23.7% in diversity compared to TD. This means that MOAD may bring from 2 to 3 more artists from different countries than TD.

	Artist	England	USA	Sweden	Iceland	Canada	Italia	India	Mexico	Norway
UBCF	Pink Floyd	X								
	In Flames			X						
	Dream Theater		X							
	Iron Maiden	X								
	Megadeth		X							
	Coldplay	X								
	Björk				X					
	The Beatles	X								
	The Cure	X								
	Motörhead	X								
TD	Pink Floyd	X								
	Björk				X					
	Johnny Cash		X							
	In Flames			X						
	Clint Mansell	X								
	Zoë Keating					X				
	Dream Theater		X							
	Iron Maiden	X								
	Megadeth		X							
	Coldplay	X								
MOAD	Films of Colour	X								
	Ondskapt			X						
	Beautiful Sin									X
	I Ribelli						X			
	Planes Mistaken for Stars		X							
	Banda Machos								X	
	Jeff Healey					X				
	Cole Swindell		X							
	Mubarak Begum							X		
	Fuck Buttons	X								

Table 2. Top-10 recommendations for a real Last.fm user

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed MOAD, an approach for music recommendations that are at the same time diverse, regarding certain aspects, and similar to user preferences concerning other aspects. We cast this problem as a multi-objective optimization task and use an efficient algorithm based on evolutionary algorithms for solving it. We have defined specific similarity functions for each considered aspect and performed several simulations using real world data to assess MOAD performance. We have compared MOAD to other well known baselines from the literature and show that it provides better results in all evaluated scenarios.

As future work, we intend to run MOAD in all possible combinations of aspects to diversify and to hold constant and with more generations. We also intend to perform an online experiment with real users. Finally, we intend to approach the same problem under the perspective of MIR replacing the user's history by a set of input artists, allowing the user to discover new artists based on her instantaneous information needs.

Acknowledgement: This work was partially funded by the EU-BR BigSea project (MCTI/RNP 3rd Coordinated Call).

8. REFERENCES

[1] Shyam Boriah, Varun Chandola, and Vipin Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 243–254. SIAM, 2008.

[2] Rocío L Cecchini, Carlos M Lorenzetti, Ana G Maguitman, and Nélida B Brignole. Multiobjective evolutionary algorithms for context-based search. *Journal of the Association for Information Science and Technology*, 61(6):1258–1274, 2010.

[3] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[4] Laizhong Cui, Peng Ou, Xianghua Fu, Zhenkun Wen, and Nan Lu. A novel multi-objective evolutionary algorithm for recommendation systems. *Journal of Parallel and Distributed Computing*, 2016.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[6] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. Adaptive diversity in recommender systems. In *IIR*, 2015.

[7] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A Konstan, and Paul Schrater. I like to explore sometimes: Adapting to dynamic user novelty preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 19–26. ACM, 2015.

[8] Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V Çatalyürek. Diversified recommendation on graphs: pitfalls, measures, and algorithms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 715–726. ACM, 2013.

[9] Ali Ouni, Raula Gaikovina Kula, Marouane Kessentini, Takashi Ishio, Daniel M German, and Katsuro Inoue. Search-based software library recommendation using multi-objective optimization. *Information and Software Technology*, 83:55–75, 2017.

[10] Elias Pampalk and Masataka Goto. Musicsun: A new approach to artist recommendation. In *ISMIR*, pages 101–104, 2007.

[11] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 19–26. ACM, 2012.

[12] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. Multiobjective pareto-efficient approaches for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):53, 2015.

[13] Markus Schedl. The lfm-1b dataset for music retrieval and recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, pages 103–110, New York, NY, USA, 2016. ACM.

[14] Saul Vargas, Pablo Castells, and David Vallet. Intent-oriented diversity in recommender systems. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1211–1212, New York, NY, USA, 2011. ACM.

[15] Saúl Vargas, Pablo Castells, and David Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 75–84, New York, NY, USA, 2012. ACM.

[16] Shanfeng Wang, Maoguo Gong, Haoliang Li, and Junwei Yang. Multi-objective optimization for long tail recommendation. *Knowledge-Based Systems*, 104:145–155, 2016.

[17] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130. ACM, 2008.

[18] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. Learning for search result diversification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 293–302. ACM, 2014.

[19] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.

[20] Yi Zuo, Maoguo Gong, Jiulin Zeng, Lijia Ma, and Licheng Jiao. Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. *IEEE Computational Intelligence Magazine*, 10(1):52–62, 2015.