Sequential Music Recommendations for Groups by Balancing User Satisfaction

Auste Piliponyte, Francesco Ricci and Julian Koschwitz

Free University of Bozen–Bolzano, Italy
{fricci,julian.koschwitz}@unibz.com

Abstract: Generating a sequence of music tracks recommendations to a group of users can be addressed by balancing the users' satisfaction for a set of recommendations (the playlist), rather than finding items that individually provide good average satisfaction to the users. In this paper we introduce a 'Balancing' technique that builds a tracks' sequence iteratively while constantly balancing users' satisfaction levels. In a live user study we have shown that it produces playlist recommendations that are better than those generated by the average preference aggregation method and comparable to those manually compiled by the group members.

1 Introduction

Group recommender systems aim at recommending the right items to a group of people in a specific occasion. One of the major issues is to satisfy the group as a whole, in an appropriate way, on the basis of the individual preferences [6][2]. Especially in the field of music the taste and preferences of individual persons are diverse and widespread. One song can never satisfy every member of the group equally. But, groups often listen to a sequence of music tracks, and this opens a new recommendation problem but also an opportunity for satisfying individual preferences [5]. While one single track may not be liked by all, a sequence of recommended songs may contain different subsets of items which are of relevance for the various members. To tackle these issues we propose a sequential recommendation technique for groups based on 'Balancing': it builds a tracks' sequence iteratively while constantly balancing user satisfaction levels. We show that this approach generally outperforms a "non-balancing" and popular technique such as 'recommendations aggregation with average'. We have implemented Balancing in a web–based music recommender and tested it in a live user study. 'Balancing' produces playlist recommendations that are better than those produced by the well-known 'Average' preference aggregation method and comparable to those manually generated by users.

2 Related Work

Apart from extensive research in the field of sequential recommendations for single user, e.g. automatic playlist generation based on track similarity [8] there has so far been significantly less effort in the area of sequential recommendations for groups [2] [9]. Masthoff [4] [5] has conducted a substantial amount of user studies in this domain. In the research with a group of people watching TV-News she observed that people, when making group recommendations, often prefer certain group rating aggregation strategies, i.e., Average, Average without Misery and Least Misery. Generally Masthoff stresses that groups care primarily about fairness within the group and stir towards “preventing misery and starvation” [5]. Having this in mind we have conjectured that for group recommendation tasks where the group consumes several recommendations (e.g., in a sequence) the ‘Balancing’ strategy, which is mentioned in the previous section, can be very promising. Baccigalupo [1] has implemented a web radio that takes into account its listeners’ preferences and plays a sequence of music. This music sequence is built iteratively by a Case-Based Reasoning process that has three major steps: Retrieve, Reuse, and Revise. In the Retrieve step they obtain a ranked list of songs. The list is produced from the entire collection of music tracks removing the tracks of recently played artists. The songs in the list are ranked according to the smoothness of the transition they would make from the previous song in the sequence. In the Reuse step the best scoring music tracks in the candidate list are re-ranked. In order to combine individual track ratings of each listener into a group rating they use a method they call satisfaction-weighted aggregation. When combining individual preferences more weight is given to the less satisfied listeners. From a newly produced ranked list they then remove the tracks that at least one listener rated below a certain misery threshold. In the final Revise step the listeners are given a possibility to adjust their preferences through explicit feedback. At the end of this step the top ranked candidate is selected and added to the music sequence.

3 Music Compilation Recommendation

In this section we present our original approach to generate a sequence of music tracks recommendations for a group of users. The technique that we propose builds a tracks’ sequence iteratively while constantly balancing user satisfaction levels. We hypothesized that our approach could produce recommendations that outperform the current state of the art techniques. Moreover, we assumed that our system would be able to compete with humans at least with respect to some aspects, such as recommendation goodness and fairness. A recommendation can be considered to be good if it satisfies each group member, and it is fair if the accumulated satisfaction level (the overall satisfaction level as it is measured so far) of each group member is similar to that of other group members. We have also made a hypothesis [7] that emotional decay is of importance when calculating cumulative satisfaction. Emotional decay describes the fading of emotions over time, which is based on the belief that user satisfaction (or dissatisfaction) with experienced items fades over time, and that items that were experienced more recently contribute more to the overall user satisfaction with a sequence of items. We have designed and developed a web application that provides music track recommendations for groups. Music track recommendations can be either produced by humans (other group members) or by the system. System recommendations are made in two major steps. In the first step the system makes single user rating predictions for each group member and for each music track. Rating predictions are produced using Matrix Factorization collaborative filtering [3]. In the second step individual recommendations are aggregated and a sequence of 10 tracks is composed and returned as the system recommendation to the group. Aggregation is done using one of three alternative aggregation approaches that are described below. The first approach is using the 'Average' of the predicted ratings to select the items to include in the playlist. First it computes the group score for each music track $i \in I$ using the formula:

$$score(G,i) = \frac{\sum_{u \in G} r^*(u,i)}{|G|}$$

Here *r*(u, i)* is the predicted rating of user *u* for item *i*, and *G* is a group user *u* belongs to. Then the ten tracks with the highest group score are returned as recommended playlist.

The second approach, 'Balancing without Decay', operates in two steps. First a candidate set is built using average aggregation, i.e., a set of candidate tracks with large average predicted rating is found. In the second step the sequence to be recommended is built using only tracks from the candidate set. While building the sequence we monitor the accumulated predicted satisfaction level of each user, i.e., the sum of the predicted ratings of the tracks. Here we assume that the user-accumulated satisfaction is equally influenced by all the previous tracks in the playlist. The accumulated satisfaction function looks as follows:

$$sat(u,S) = \frac{\sum_{i \in S} r^*(u,i)}{\sum_{i \in M} r^*(u,i)}$$

Where *u* is a user, *i* is a track, *r*(u,i)* is the predicted rating for track *i* and user *u*. If *u* has rated *i*, then the true rating is used. *S* is track sequence that has been built till that moment. *M* is a set of |*S*| tracks that have the highest explicit or predicted rating for user *u* in the entire collection. The set *M* is the set that would be recommended to the user if he had requested an individual recommendation and it is used to normalize the user satisfaction. In order to select a new track to be added to a partially completed recommendation sequence we calculate for each remaining track in the candidate list the accumulated satisfaction of each group member with the sequence that would be produced after adding that track to the current sequence. Having done that, we calculate sums of all possible differences between the group members' satisfactions. Finally, we select and add the track that has the smallest sum of satisfaction differences.

This process is iterated, starting with a sequence of one single track (having the largest average satisfaction) until a sequence of desire length (10 tracks in our experiments) is obtained. This is finally recommended to the user.

The third aggregation approach is 'Balancing with Decay' which differs from 'Balancing without Decay' approach only with respect to the cumulative satisfaction function used. In 'Balancing with Decay' approach user cumulative satisfaction is calculated using the following formula:

$$sat(u,S) = \frac{\sum_{k=1}^{|S|} \gamma^{|S|-k} \; r^*(u,i_k)}{\sum_{l=1}^{|M|} \gamma^{|M|-k} \; r^*(u,i_l)}$$

Where *S* is track sequence that has been built till that moment. *M* is a set of |*S*| tracks that have the highest explicit or predicted rating by user u in the entire collection, *u* is a user, *ik* is a track from *S*, while *il* is a track from *M*. *r*(u,i)* is the predicted rating for track *i* and user *u*. If *u* has rated *i*, then the true rating is used. Finally, γ is a decay parameter. The decay parameter ensures that recent tracks get more importance when calculating user satisfaction. In order to test our hypotheses we implemented a system that enables users to enter ratings; set playlist recommendations for groups composed by a master user; evaluate playlist recommendations built by the system with the three mentioned approaches and those generated by the group members. The total number of users that have registered and left at least one music track rating in the study was 77. Users have left 5160 ratings in total with the average of 67 ratings per user. With 1068 music tracks in our dataset, this amounts to a 6% density of the ratings. When compared to the density of standard recommender system datasets (Netflix Challenge dataset: 1.17%; Yahoo! Music dataset: 0.04%), it can be considered as a not sparse data set. At the beginning of the experiment it was necessary for each participant to rate a substantial amount of music tracks. Then, participants were divided into groups, which were composed automatically by building a group of three users as soon as three new members registered to the system. The users were requested to make music track sequence recommendations for their groups. In order to accomplish that task users were able to browse the ratings of the other group members for assessing their music preferences. Afterwards they were presented with two sequences, a system recommendation, that was built using one of the three methods mentioned above, and a track sequence produced by a randomly chosen group member, through a set of questions they had to evaluate both sequences in comparison (Fig. 1). Users were not aware of who had generated the recommendations. We provide bellow a short summary of the results since the complete results would extend the scope of this workshop paper, for more information the reader is referred to [7]. Testing 'Average' algorithm against 'Balancing', the 'Balancing with Decay' method was more often preferred to human recommendations than the 'Average' method was. When users had to choose playlists created with 'Average' in comparison to user-generated playlists, 72% selected the latter. 'Balancing without Decay' gained better results with 58% of the participants selecting user generated playlist and 'Balancing with Decay ' scored a solid 62% in favor of the computed aggregation method opposed to 38% for the user generated playlists.

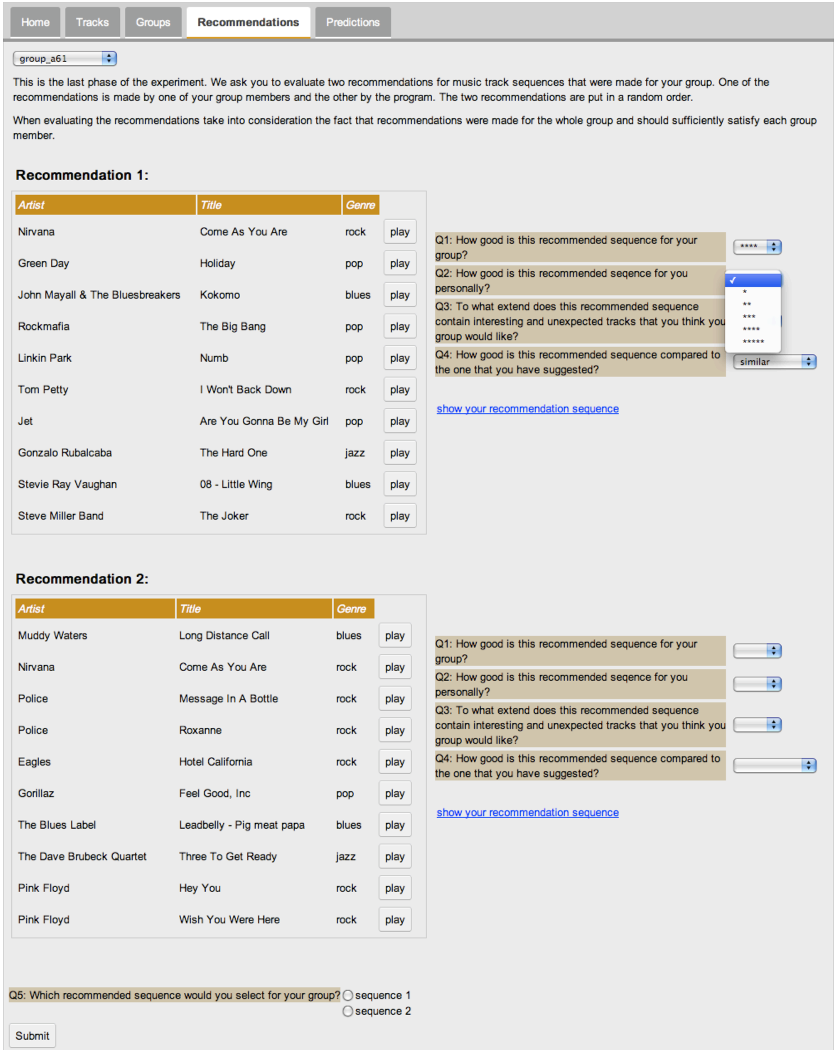

Fig. 1. Music track evaluation

So we can conclude that 'Balancing' can achieve better performance than 'Average' algorithm. Moreover, its performance is of comparable quality to humans. This is remarkable, because we have provided users with effective tools for the construction of group recommendations and users spent a considerable amount of time in building these playlist recommendations. In addition to that, it should not be forgotten, that the main purpose of recommender systems is automatizing the recommendation process. Making recommendations is a laborious and demanding activity. This is especially true for group recommendations, where preferences of multiple people have to be combined. Therefore, 'Balancing' approach could be considered as a good alternative for human recommendations.

4 Future Work

We note here that sequential recommendations is an interesting research area and this type of problems are naturally generated by decision making activities in groups. Our long term goal is to develop computational solutions to sequential recommendation problems even further and specifically we aim at what we call “stable groups”, i.e., groups that have a persistent state, which receive several recommendations at different points in time, and therefore can be the target for sequential recommenders. Hence, our aim on a long run is to develop this approach further and specifically aim at so called stable groups that we see as the main target for sequential recommenders because of the nature of their composition. With stable we mean a group that stays over a long time in the same formation, like a family, colleagues at a workplace or groups of friends. Apart from music recommendations we plan to examine other domains like collective cooking or suggesting sports activities in order to achieve a conscious diet and a healthy lifestyle.

References

1. Baccigalupo, C. Poolcasting– An Intelligent Technique to Customise Music Programmes for Their Audience. PhD Thesis, UAB (2009)
2. Jameson A., Smyth B., Recommendation to groups. In *The adaptive web*, Brusilovsky P., Kobsa A., and Nejdl W. (Eds.). Springer-Verlag, Berlin, Heidelberg 596-627 (2007)
3. Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. KDD ’08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434, New York, NY, USA, ACM. (2008)
4. Masthoff, J. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *UMUAI,* volume 14, pp. 37-85 (2004)
5. Masthoff, J. Group recommender systems: Combining individual models. In Ricci, F., Rokach, L., Shapira, B., Kantor, P. (Eds.), Recommender Systems Handbook (pp. 677-702). Dordrecht, Netherlands: Springer Science+Business Media. Pre-final version. Springer-Verlag (2011)
6. McCarthy K, Mcginty L, Smyth B. Case-based group recommendation: compromising for success. In: Proceedings of the 7th international conference on case-based reasoning: case-based reasoning research and development. pp. 299–313, Springer, Berlin (2007)
7. Piliponyte, Auste. Sequential Group Recommendations. MA Thesis, Free University of Bozen–Bolzano. (2012)
8. Ragno, R., Burges, C., and Herley, C. Inferring similarity between music objects with application to playlist generation. *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval* (MIR '05). ACM, New York, NY, USA, pages 73-80 (2005)
9. Salamó M., McCarthy K., Smyth B., Generating recommendations for consensus negotiation in group personalization services. Personal and Ubiquitous Computing 16(5): 597-610 (2012)