

Evaluating Music Recommender Systems for Groups

Zsolt Mezei
Dept. of Computer Science
ETH Zürich
Zürich, Switzerland
mezei3zsolt@yahoo.com

Carsten Eickhoff
Dept. of Computer Science
ETH Zürich
Zürich, Switzerland
ecarsten@inf.ethz.ch

ABSTRACT

Recommendation to groups of users is a challenging and currently only passingly studied task. Especially the evaluation aspect often appears *ad-hoc* and instead of truly evaluating on groups of users, synthesises groups by merging individual preferences.

In this paper, we present a user study, recording the individual and shared preferences of actual groups of participants, resulting in a robust, standardized evaluation benchmark. Using this benchmarking dataset, that we share with the research community, we compare the respective performance of a wide range of music group recommendation techniques proposed in the literature.

CCS CONCEPTS

•**Information systems** → *Recommender systems*; Personalization;
•**General and reference** → *Surveys and overviews*; Evaluation;

KEYWORDS

User study, evaluation, ranking, music, groups, recommendations.

1 INTRODUCTION

Recommender systems are widely used both in industrial and academic settings. E-commerce platforms and content providers use them to recommend new products in order to maintain frequent interaction with their service. Most of these systems focus on individual users as the target of their recommendations (e.g., [1, 5, 8, 11]).

In turn, group recommendation is a new task in which an item is recommended to a group of users who will consume it together. Such a feature is useful for streaming movie or music providers when recommending content to a family or a group of friends who want to watch a film together. The same holds for search queries, travel offers or any type of products to be bought or consumed together. The challenge in such a task is modeling and matching the taste of the entire group even though only individual preferences may be known. There are various alternative proposals in the academic literature regarding how to combine the preferences of individuals in some way to generate the group taste [2, 9, 10, 13, 15].

As a consequence of the transient nature of groups, virtually all state-of-the-art music recommendation systems for groups evaluate their performance in an *ad-hoc* manner on the basis of synthesised groups sampled from individuals that never actually interacted or formed a group. While this approach is easily implemented, it also naïvely ignores true group dynamics and makes for a sub-optimal benchmarking setup.

In this paper, we take a different approach by collecting true group preference data through a user study described in Section 2. To the best of our knowledge, this dataset represents the first resource of its kind. Finally, on the basis of our collection, Section 4 compares a wide range of existing music group recommendation systems as well as basic machine learning algorithms.

2 SURVEY PROTOCOL

To collect a dataset of true expressions of group preferences, we conduct a supervised user study in which we team up participants in groups of three people, and, for a duration of approximately 15 minutes have them listen to music while engaging in a social activity (playing the card game Uno²). While doing so, they rate the various tracks that are played to them as a group, rather than individually.

The study is facilitated via a smartphone-centered Web application. After logging in to the study and providing basic demographic information (age and gender) the participants answer ten personality test questions³ with two possible answer choices each. Finally, they are asked to select at least 5 favorite tracks out of a larger song database.

Once this basic information is recorded, the participants begin playing their card game while a shuffled random selection of 10 of their pooled favorite song choices is being played. In order to account for track length variability, each song was faded out after the first 1.5 minutes. At each song change, the mobile phone survey application prompts participants to individually rate the current track on a 5-point Likert scale, following the 1 (lowest) – 5 (highest) star rating scheme employed by many popular content providers.

At the end of the experiment, the participants are asked to rate all played songs once more, as a group, needing to form an agreement between their individual preferences. During the entirety of the study, an experimenter is present in the room in order to take note of social interaction not recorded by the survey application. All subjects were recruited at and around a major university campus and were compensated for their participation.

3 DATASET STATISTICS

We conducted 26 experiments, each involving three participants, resulting in an overall pool of 78 unique subjects. In total, 1068 user and 356 group ratings for songs were recorded. The average user rating is 3.36 with standard deviation 1.27, while the average among groups is 3.3 with a tighter standard deviation of 1.08. Groups can be noted to be more reserved and choose the far ends of the rating scale less frequently than individual users.

Presented at the 2017 Workshop on Value-Aware and Multistakeholder Recommendation.

²[https://en.wikipedia.org/wiki/Uno_\(card_game\)](https://en.wikipedia.org/wiki/Uno_(card_game))

³<https://www.41q.com/>

3.1 Songs

The song database consists of 100 tracks and was populated with recently popular⁴, as well as all-time favorite tracks⁵. An additional number of songs were added manually in order to increase diversity in terms of music styles.

Each track is annotated in terms of metadata information including the title, artists, album and genre. The list contains tracks by 87 different artists, 82 albums and 74 distinct genres. In case of 11 tracks information regarding the album is missing, and 4 tracks do not have genre tags. Wave-form acoustic similarities between pairs of songs are computed using the Marsyas⁶ sound analysis library as proposed by Sprague *et al.* [18].

There are 17 songs that were never chosen by any participant of the study. On average an individual elected a song 3.99 times with a standard deviation of 3.98. Multiple users in a group could choose the same songs, on average a song is present 3.56 times in a group's playlist with standard deviation of 3.30. The minimum size of a group's global playlist before sampling was 11 and the maximum 17, with an average of 13.69 and a standard deviation of 1.59.

3.2 Qualitative Findings

When asked for open-ended feedback after study completion, participants stated that they enjoyed the combination of a social activity and listening to music in order to make the study immersive. This frequently resulted in participants wanting to play out the last game of cards when the survey duration ended. No participant reported difficulties operating the user interface or switching contexts from playing the game to rating songs when tracks changed.

Participants enjoyed the group setup, and shared with the others sentiments like "That's my song!" or "Who chose that one? It's terrible." A few subjects had strong feelings linked to some of the songs, and it was important to them to influence the final group vote for those, while they did not care about most other tracks' ratings. Participant personality played a significant role in the observed interaction. While some of them were shy and did not actively join the discussion or debate, others were consistently more active/dominant. In one case, a subject individually gave 5 stars for a song, but during the group voting phase agreed when the group assigned only 3, without mentioning that he/she actually liked the track. When two subjects shared the same taste and rated songs in a similar manner, we noted them to suppress the third member's opinion in the discussion. During a few sessions, participants considered their performance in the game when rating tracks, giving higher votes during winning, rather than losing rounds.

Some of the groups agreed to rate tracks by assigning the average among individual votes as the group rating. Despite such prior agreements, several of these groups ended up discussing individual tracks instead of averaging. There was also a single case of a group not agreeing on how to vote, but after a long discussion finally settling for the average.

4 SYSTEM COMPARISON

This section compares a wide range of methods that attempt to infer group preferences from individual votes. The previous discussion of qualitative observations of group voting behavior underlines the non-trivial nature of this task even for humans. While previous work has neglected collecting actual group preferences and instead employs a range of approximations, here we use the observed actual group vote as ground truth. We propose two evaluation metrics to assess recommendation quality. Firstly, the mean squared error (MSE) between true and inferred ratings is calculated. In an alternative evaluation approach, casting the problem as a ranking task, we order songs according to decreasing inferred group preference and compute ranking quality in terms of nDCG. The relevance of each track is equal to the observed true group preference, ranging from 5 (most relevant) to 1 (least relevant) Evaluation is performed in leave-one-out fashion, training models on 25 groups at a time and evaluating on the remaining one.

4.1 Methods from Literature

In this section, we present a comparison of 8 different algorithms from published papers. They were selected to represent the full spectrum of currently known group recommendation techniques and were implemented in Scala.

Ali *et al.* [2] introduce three recommendation methods. Besides the geometric average as well as the minimum across all individual ratings, the authors discuss a weighted averaging policy, where the weight is given by the overall number of votes collected from the user, normalised by the total number of ratings. In our scenario, all group members have exactly the same number of votes. In consequence, we calculate weights on the basis of user attributes. We define a set of attributes A such that the mean classification error is minimized. The weight is represented by the number of times a user's attribute is in the set A .

Amer-Yahia *et al.* [3] discuss a consensus function that, for user $u \in U$ and song s discounts the average rating by the disagreement variance across individual ratings $r_{s,u}$ as presented in Equation 1.

$$r_s = \lambda avg_s + (1 - \lambda) \left(1 - \frac{\sum_{u \in U} (r_{s,u} - avg_s)^2}{|U|} \right) \quad (1)$$

Dias *et al.* [7] propose a collaborative filtering approach based on latent factor spaces to recommend songs. They cluster users and choose leaders (the closest users to cluster centroids). These leaders' preferences are used to add diversity and smooth the group recommendations. The number of leading singular values were learned during the training process. The final score for a song is the weighted average of the group members and each leader. The mixture weight is the normalised number of ratings given by the user.

Chao *et al.* [6] rely on negative "preferences" to recommend music. Such negative votes are defined by a rating $r_{s,u} \leq 3$. The system randomly plays any tracks which did not receive a negative vote from any group member, and is not on the same album with any song with a negative vote.

Liu *et al.* [14] play songs randomly from the group members' playlists. We generate random numbers from a continuous uniform distribution on the interval [1, 5].

⁴<http://www.billboard.com/charts/hot-100/2016-04-16>

⁵<http://www.telegraph.co.uk/culture/music/11621427/best-songs-of-all-time.html>

⁶<http://marsyas.info/>

Table 1: Results of methods from academic literature

Method	MSE mean	MSE sd	nDCG mean	nDCG sd
[2] avg	0.593	0.299	0.942	0.048
[2] min	1.642	0.932	0.922	0.053
[2] weighted	0.637	0.344	0.937	0.059
[3]	0.596	0.299	0.941	0.052
[7]	0.635	0.284	0.938	0.049
[6]	1.316	0.459	0.865	0.030
[14]	2.578	0.387	0.777	0.038
[18]	0.586	0.299	0.938	0.053

Kukka *et al.* [12], O'Hara *et al.* [16] and Sørensen *et al.* [17] introduce different democratic voting methods. In each round the users can vote, and the highest voted song is played next. We approximate this behavior by the unweighted average across votes.

Sprague *et al.* [18] use democratic voting combined with waveform song similarity. The final ranking criterion is a weighted mixture of the rating mean and the distance-discounted ratings of other songs. In this manner, explicit votes (0 distance to the current song) bear the greatest weight, while information from other, similar tracks can be elicited in the recommendation.

$$O_{song_i} = \lambda \cdot avg_{song_i} + \frac{(1-\lambda)}{\#songs} \sum_{\substack{song_j \\ j \neq i}} (avg_{song_j} \cdot dist_{song_i, song_j}) \quad (2)$$

Table 1 shows the performance comparison between the presented systems on our dataset. [6] achieve low accuracy, as it merely divides songs into two sets: playable and not playable. The strongest overall performance, dependent on the chosen metric is observed for [2]'s unweighted averaging (lowest MSE) and [18]'s song similarity modification to this scheme (highest nDCG). We employ a Wilcoxon signed-rank test at $\alpha < 0.05$ -level and find these leading methods to be statistically indistinguishable. The remaining field of methods shows significantly lower recommendation performance.

4.2 Machine Learning Algorithms

In addition to the domain-specific group recommendation techniques presented earlier, this section introduces a range of traditional machine learning as well as deep neural network techniques. All algorithms are implemented in Scala using the Spark Machine Learning Library⁷ or Python and Tensorflow⁸, respectively.

The classifiers attempt to predict observed group votes on the basis of a range of features: user attributes (demographics and personality questions) and individual ratings, average of all group members' votes, disagreement variance calculated in Equation 1, minimum and maximum of the members' ratings, global average of all users' votes across songs, music metadata (title, artist, album, genre information), song similarity presented in [18].

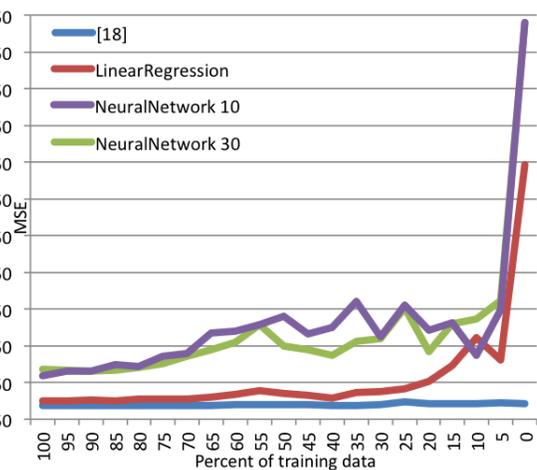
We investigated various feature subsets and combinations, finding linear regression to consistently show the lowest MSE, resulting in a performance comparable to that of the best previously described dedicated group recommendation schemes. Table 2 gives a comprehensive overview of results. Using the same feature set, Figure 2

⁷<http://spark.apache.org/docs/latest/ml-guide.html>

⁸<https://www.tensorflow.org/>

Table 2: Results of machine learning algorithms

Method	MSE mean	MSE sd	nDCG mean	nDCG sd
LinearRegression	0.599	0.306	0.942	0.056
Lasso	1.194	0.393	0.851	0.051
RidgeRegression	0.779	0.307	0.935	0.052
LogisticRegression	0.881	0.393	0.889	0.053
DecisionTreeClas	0.782	0.383	0.910	0.052
DecisionTreeReg	0.623	0.295	0.938	0.050
NaiveBayes	1.312	0.653	0.802	0.040
RandomForestClas	0.663	0.312	0.919	0.056
RandomForestReg	0.612	0.286	0.944	0.044
GradientTreeReg	0.626	0.297	0.937	0.050
NeuralNetwork 10	0.668	0.274	0.910	0.057
NeuralNetwork 30	0.686	0.285	0.903	0.063
NeuralNetwork 60,40	0.670	0.299	0.922	0.051

**Figure 1: Recommendation performance of [18], linear regression, neural network as a function of training set size.**

shows the architecture of the neural network with 10 hidden units, where the output is a rating from 1 to 5. All permutations of the group members' ratings (Rating1, Rating2, Rating3) are used during training.

Figure 1 shows the results of [18], a linear regression classifier with SGD Optimization, and a feed-forward neural network as we vary the amount of available training data. The experiment suggests that further gains could be achieved if more training data had been available. Linear regression and neural network algorithms show a more systematic benefit from additional data than did the group recommendation schemes analyzed before. Although some configurations of these supervised classifiers were able to match or mildly outperform the previously strongest methods, the limited dataset size did not allow us to confirm statistical significance of such differences.

5 CONCLUSION

In this paper, we describe a user study and benchmarking dataset in which natural interactions of 78 participants with music recommendation systems for groups are elicited. The goal of this study is to

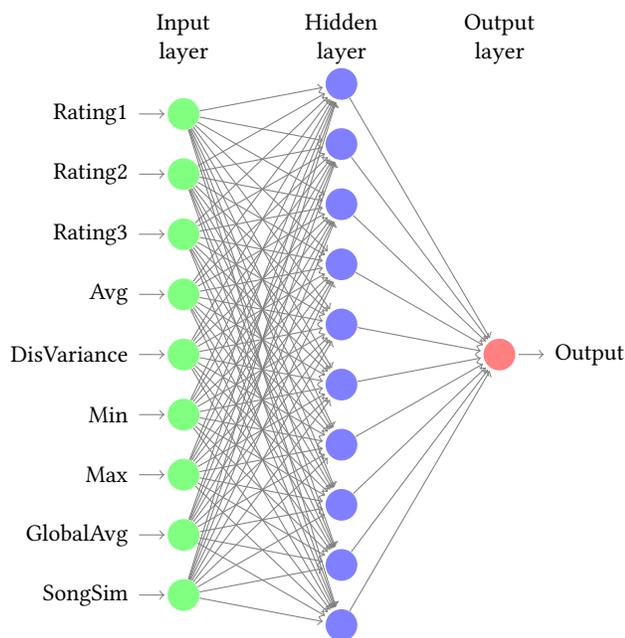


Figure 2: Neural Network with 10 hidden units

provide a robust testbed for evaluation of group recommendation systems. Demonstrating the merit of this resource, we evaluate a wide range of state-of-the-art music recommendation systems for groups as well as general-purpose machine learning methods on the task of inferring group preferences from individual user votes.

Among the compared methods, despite the presence of a few local optima, unweighted averaging of individual votes has been confirmed as the most robust and generally applicable choice of recommendation scheme.

This study investigated fixed group sizes of three participants in order to limit inter-experiment variance. In the future, it would be interesting to study different group sizes as well as the behavior of organic (e.g., friends, families, etc.) versus randomly assigned groups. On the recommender systems side, it would be interesting to investigate playlist generation for groups (rather than recommending individual tracks as studied here). Besides meeting the group's preferences, such playlists need to satisfy constraints such as continuity, diversity or similarity [4].

REFERENCES

- [1] Natalie Aizenberg, Yehuda Koren, and Oren Somekh. 2012. Build Your Own Music Recommender by Modeling Internet Radio Streams. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*. ACM, New York, NY, USA, 1–10. DOI: <http://dx.doi.org/10.1145/2187836.2187838>
- [2] Irfan Ali and Sang-Wook Kim. 2015. Group Recommendations: Approaches and Evaluation. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (IMCOM '15)*. ACM, New York, NY, USA, Article 105, 6 pages. DOI: <http://dx.doi.org/10.1145/2701126.2701208>
- [3] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group Recommendation: Semantics and Efficiency. *Proc. VLDB Endow.* 2, 1 (Aug. 2009), 754–765. DOI: <http://dx.doi.org/10.14778/1687627.1687713>
- [4] Geoffroy Bonnin and Dietmar Jannach. 2014. Automated Generation of Music Playlists: Survey and Experiments. *ACM Comput. Surv.* 47, 2, Article 26 (Nov. 2014), 35 pages. DOI: <http://dx.doi.org/10.1145/2652481>
- [5] Badrish Chandramouli, Justin J. Levandoski, Ahmed Eldawy, and Mohamed F. Mokbel. 2011. StreamRec: A Real-time Recommender System. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*. ACM, New York, NY, USA, 1243–1246. DOI: <http://dx.doi.org/10.1145/1989323.1989465>
- [6] Dennis L. Chao, Justin Balthrop, and Stephanie Forrest. 2005. Adaptive Radio: Achieving Consensus Using Negative Preferences. In *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (GROUP '05)*. ACM, New York, NY, USA, 120–123. DOI: <http://dx.doi.org/10.1145/1099203.1099224>
- [7] Pedro Dias and João Magalhães. 2013. Music Recommendations for Groups of Users. In *Proceedings of the 2013 ACM International Workshop on Immersive Media Experiences (ImmersiveMe '13)*. ACM, New York, NY, USA, 21–24. DOI: <http://dx.doi.org/10.1145/2512142.2512151>
- [8] Carlos A. Gomez-Uribe and Neil Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manag. Inf. Syst.* 6, 4, Article 13 (Dec. 2015), 19 pages. DOI: <http://dx.doi.org/10.1145/2843948>
- [9] Anthony Jameson. 2004. More Than the Sum of Its Members: Challenges for Group Recommender Systems. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 48–54. DOI: <http://dx.doi.org/10.1145/989863.989869>
- [10] Anthony Jameson and Barry Smyth. 2007. *The Adaptive Web*. Springer-Verlag, Berlin, Heidelberg, Chapter Recommendation to Groups, 596–627. <http://dl.acm.org/citation.cfm?id=1768197.1768221>
- [11] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, New York, NY, USA, 165–172. DOI: <http://dx.doi.org/10.1145/2043932.2043964>
- [12] Hannu Kukka, Rodolfo Patino, and Timo Ojala. 2009. UbiRockMachine: A Multimodal Music Voting Service for Shared Urban Spaces. In *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia (MUM '09)*. ACM, New York, NY, USA, Article 9, 8 pages. DOI: <http://dx.doi.org/10.1145/1658550.1658559>
- [13] Henry Lieberman, Neil W. Van Dyke, and Adrian S. Vivacqua. 1999. Let's Browse: A Collaborative Web Browsing Agent. In *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI '99)*. ACM, New York, NY, USA, 65–68. DOI: <http://dx.doi.org/10.1145/291080.291092>
- [14] KuanTing Liu and Roger Andersson Reimer. 2008. Social Playlist: Enabling Touch Points and Enriching Ongoing Relationships Through Collaborative Mobile Music Listening. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. ACM, New York, NY, USA, 403–406. DOI: <http://dx.doi.org/10.1145/1409240.1409299>
- [15] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A Recommender System for Groups of Users. In *Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work (ECSCW'01)*. Kluwer Academic Publishers, Norwell, MA, USA, 199–218. <http://dl.acm.org/citation.cfm?id=1241867.1241878>
- [16] Kenton O'Hara, Matthew Lipson, Marcel Jansen, Axel Unger, Huw Jeffries, and Peter Macer. 2004. Jukola: Democratic Music Choice in a Public Space. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '04)*. ACM, New York, NY, USA, 145–154. DOI: <http://dx.doi.org/10.1145/1013115.1013136>
- [17] Henrik Sørensen and Jesper Kjeldskov. 2012. Distributed Interaction: A Multi-device, Multi-user Music Experience. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 336–339. DOI: <http://dx.doi.org/10.1145/2254556.2254621>
- [18] David Sprague, Fuqu Wu, and Melanie Tory. 2008. Music Selection Using the PartyVote Democratic Jukebox. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 433–436. DOI: <http://dx.doi.org/10.1145/1385569.1385652>