# Haven't I just Listened to This?: Exploring Diversity in Music Recommendations

Antonela Tommasel
antonela.tommasel@isistan.unicen.edu.ar
ISISTAN (CONICET/UNCPBA)
Tandil, Argentina

Juan Manuel Rodriguez
juanmanuel.rodriguez@isistan.unicen.edu.ar
ISISTAN (CONICET/UNCPBA)
Tandil, Argentina

Daniela Godoy
daniela.godoy@isistan.unicen.edu.ar
ISISTAN (CONICET/UNCPBA)
Tandil, Argentina

## ABSTRACT

Recommender systems have recently been criticized for promoting bias and trapping users into filter bubbles. This phenomenon not only limits potential user interactions but also threatens the broadness of content consumption. In a music recommender, for example, this situation can limit user perspective as music allows people to develop cultural knowledge and empathy. As a fundamental characteristic of users' content consumption is its diversity, it is necessary to break the bubbles and recommend potentially relevant and diverse songs from outside the influence of such bubbles. To address this problem, we present `MRecuri` (Music RECommender for filter bUbble diveRsIfication), a music recommendation technique to foster the diversity and novelty of recommendations. A preliminary evaluation over Last.fm listening data showed the potential of `MRecuri` to increase the diversity and novelty of recommendations compared with state-of-the-art techniques.

## CCS CONCEPTS

• **Information systems → Social recommendation**; • **Computing methodologies** → *Neural networks.*

## KEYWORDS

Recommender systems, Filter bubbles, Music recommendation, Diversity

**ACM Reference Format:**
Antonela Tommasel, Juan Manuel Rodriguez, and Daniela Godoy. 2022. Haven't I just Listened to This?: Exploring Diversity in Music Recommendations. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '22 Adjunct), July 4–7, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3511047.3536409

## 1 INTRODUCTION

Music streaming services have become popular in the last few years, contributing to the democratization of music access [11]. In most cases, recommender systems provide users with personalized items similar to the content they have previously indicated an interest. While this approach might help increase click rate, sales, or conversion rates, it does not necessarily induce users to explore new and diverse content [9]. Users with little or no exposure to diverse (or even contradicting) views can become unintendedly trapped in an isolated filter bubble [9, 23]. Filter bubbles complement echo chambers as both result in users consuming content aligned with their views. In both cases, recommenders are induced to narrow their suggestions, reinforcing user segregation and other biases [21].

While music access in streaming services seems fluid and diverse, platforms have been acknowledged to recommend items in circumscribed tiers for users and listening environments in connection with social structures [16, 18, 20]. If streaming platforms foster filter bubbles, users would not be encouraged to discover music that differs from their taste, limiting their openness and cultural awareness [16, 18, 29]. The need for promoting diverse and novel content has also been acknowledged to create healthy consumption patterns and contribute to streaming platforms' success [12].

In this work, we tackle the music recommendation problem by fostering track recommendation diversification in a filter bubble setting. To this end, we present `MRecuri`, a Music RECommender for filter bUbble diveRsIfication. This technique focuses on implicitly discovering and characterizing filter bubbles based on music listening history, a music knowledge graph, and user social interactions. Then, its goal is to present users with track recommendations that challenge the similarities between the characteristics of the listened tracks, thus helping to expand users' listening horizons. To support this proposal, we conducted a preliminary evaluation over a Last.fm data collection and compared our technique with state-of-the-art recommenders. Results showed that `MRecuri` allowed recommending tracks that were likely to be relevant while being different from the already known ones, fostering the exploration of the content space and thus helping to reduce the filter bubble effect in music recommendation.

## 2 RELATED WORKS

In the music domain, recommenders aim to provide users with a set of relevant tracks or artists based on indicators such as the listening frequency, demographic user information, and acoustic features, among others [28]. Traditionally, recommenders aim at maximizing the relevance of the provided recommendations. Nonetheless, other qualities have recently started to consider the long-term effects of recommendations, such as diversity and retention [1].

Several works [6, 17, 31] have explored alternatives inspired by Maximal Marginal Relevance (MRR) [3] to adjust recommendations based on a pre-defined function considering both relevance and diversity. Vargas and Castells [31] diversified recommendations according to user sub-profiles based on item tags and then re-ranking the recommendations made for each sub-profile. Di Noia et al. [6] re-ranked recommendations based on users' diversity proneness towards different attributes (e.g., genre, actor, director). Note that if users do not show sufficient proneness, they might not get diverse


*UMAP '22 Adjunct, July 4–7, 2022, Barcelona, Spain*

ACM ISBN 978-1-4503-9232-7/22/07.
https://doi.org/10.1145/3511047.3536409

enough recommendations and thus, would remain in their filter bubble. Finally, Lu and Tintarev [17] proposed tailoring diversity importance based on users' emotional stability, thus requiring users to fill out a questionnaire or provide access to their social media to infer their personality traits, which might not be simple.

In summary, as most techniques only consider diversity, there is no guarantee that the new recommended elements will be novel. In turn, users' experiences might not be effectively broadened, contributing to strengthening filter bubbles. Most studies either ignore the existence of filter bubbles or assume their explicit existence. In the first case, techniques might inadvertently perpetuate the bubbles. In the second case, it is necessary to define the criteria for finding the bubbles, including the selected technique and the explicit relationships or interactions between nodes, which could be computationally complex. In addition, most existing approaches also ignore user context and social relations and how they affect or model their interests [7, 22]. Instead, in `MRecuri`, community structures or bubbles are implicitly induced to seamlessly adapt to changes in content consumption patterns and user interactions. This adaptive approach allows for more freedom in bubble definition and more sensitivity to changes in the network.

## 3 DIVERSIFYING MUSIC RECOMMENDATIONS

A fundamental challenge for broadening users' recommendations is how to learn the dynamic filter bubble configurations based on the consumed items and the social interactions between users. `MRecuri` is designed to implicitly learn users' consumption patterns to adapt to their particularities, aiming to strike a balance between relevance and diversity. `MRecuri` is inspired by `FRediECH` (a Friend RecommenDer for breakIng Echo CHambers) [30], which was devised as an echo chamber-aware friend recommendation approach that learns users and echo chamber representations from the shared content and past users' and communities' interactions to recommend novel and diverse users in social media.

`MRecuri`'s overall architecture is schematized in Figure 1. It takes as input the track knowledge graph, user listening history and interactions. For each user, it outputs a ranking of tracks according to their listening likelihood strength.

***User representation.*** Users usually interact with others, which can also affect their preferences. Despite traditionally associated with homophily, Pálovics and Benczúr [22] showed that users increase new content consumption after a friend listened to it, compared to users who were not socially exposed to new content. GCNs [14] allow representing nodes based on their characteristics and those of their interactions. Here, users are represented based on trainable embeddings of their latent features ($U$) and the social interaction graph. The GCN was implemented using Spektral[1]:

$$GCN = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XW + b) \quad (1)$$

Here, $\sigma$ represents a linear activation function and $A$ the adjacency user matrix (i.e., the user social interactions). Then, $\hat{A} = A + I$ and $\hat{D_{ii}} = \sum_j \hat{A_{ij}}$, where $\hat{D}$ is a diagonal matrix. The goal of $\hat{D}$ is to act as a normalizer of $\hat{A}$ to avoid weights linearly scaling as the number of user interactions increases. $X$ represents the defined user embeddings, $W$ is a matrix of trainable weights, and $b$ is the trainable bias vector. The output of this GCN represents users and their interactions, and it is concatenated with the user embeddings to generate the intermediate user representation. This concatenation aims at better learning each user's particularities, regardless of their social connections, while facilitating the training process.

***Track representation.*** In addition to acoustic or audio features, music tracks are inherently related [11, 12]. Inspired by Castellano et al. [4], we defined a music knowledge graph to represent tracks and artists. Knowledge graphs have emerged as abstractions to organize structured knowledge and integrate different types of information. Tracks are connected to tags, while artists are connected to tracks and tags. In the future, other relations could be included.

Tracks are represented by their trainable embeddings, 11 features extracted from Spotify (energy, instrumentalness, liveness, speechniness, acousticness, danceability, valence, mode, key, loudness, and tempo), and the artist and average tags embedding. These embeddings and the Spotify features are concatenated to generate the intermediate track representation. Despite tracks, tags and artists relations were extracted from the built knowledge graph, they are not represented by a graph as artists and tags might be shared by multiple tracks, and thus the graph would be dense, increasing the computational complexity (and the need for hardware resources).

***Making recommendations.*** The prediction model is inspired by the Deep & Wide architecture and aims to predict the likelihood of a particular user-track combination. The Deep part aims to better generalize over the unseen interactions, while the Wide part aims to better learn the implicit characteristics of users.

The output of the Deep part is obtained by concatenating user and track representations and passing them through 3 dense layers with linear activation functions. These multiple layers allow the model to learn non-linear relations between users and tracks. For the Wide part, user, track, artist, and average tag embeddings are combined and passed through a dense layer. The *combine* operation (green boxes) concatenates users and tracks. Given user matrix $U \in \mathbb{R}^{n \times m_1}$ where $n$ is the number of users and $m_1$ the size of user embeddings, and track matrix $T \in \mathbb{R}^{t \times m_2}$ where $t$ is the number of tracks, and $m_2$ the size of track embeddings, the combined matrix belongs to $\mathbb{R}^{(n \times t) x (m_1 + m_2)}$ whose rows are the concatenation of each row of $U$ with each row of $T$. Despite this allows estimating all listening likelihood strengths simultaneously, due to hardware limitations, estimations are made for a user-track pair at the time.

Finally, the estimated listening strength is obtained by adding the Wide and Deep outputs. This strength can also be seen as the estimation of the times a user would listen to a track (referred to as *scrobbles*[2]). It is assumed that the higher the strength, the better the balance between the relevance of recommendations and their distance to users' filter bubbles.

***Model training.*** We defined a loss function (Eq. 2) based on the distance between users and tracks, where $Y_{ut}$ is the number of scrobbles of user $u$ for track $t$. Since 3% of scrobbles differed in at least one magnitude order from the others, they were set to the maximum number of the other 97%. $\hat{Y}_{ut}$ represents the predictions, $|E|$ the number of predictions, and $d\,(u, t)$ represents the distance

[1] https://graphneural.network/

[2] This term comes from *scrobbling*, which represents the act of logging the songs that a user has listened to.

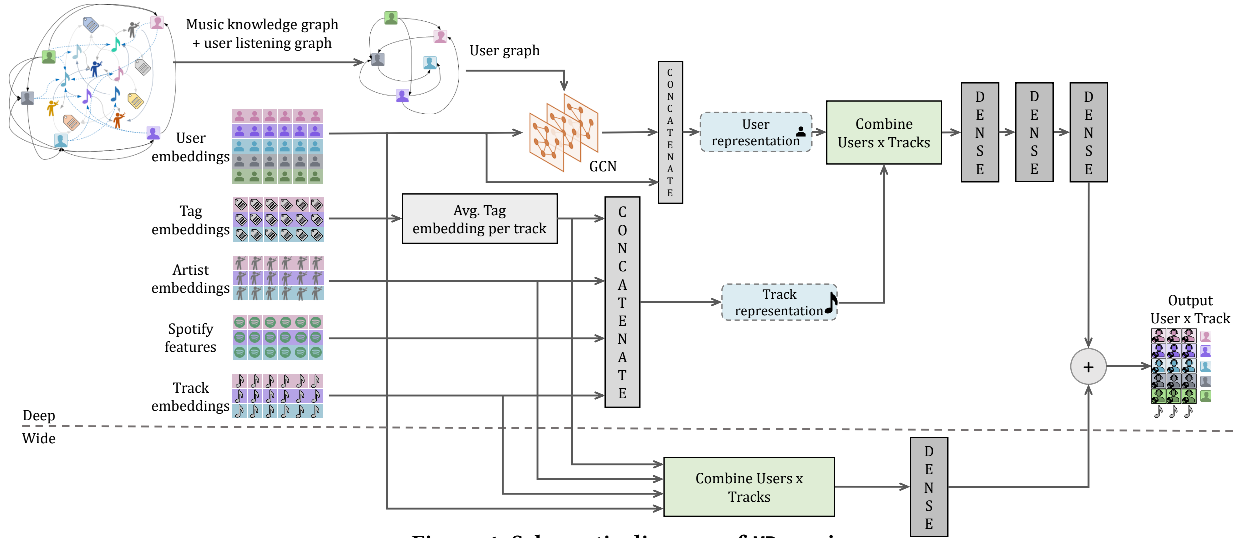


**Figure 1: Schematic diagram of MRecuri**

between user $u$ and track $t$. As scrobbles might span over an extensive range, the logarithm helps to scale their values closer to zero, which facilitates learning and helps to avoid exploding gradients. Then, for negative sampling, scrobbles were multiplied by 2 to avoid a prediction of 0 when having a scrobble of 1.

$$L\left(Y, \hat{Y}\right) = \frac{\sum d\,(u,t)^{\beta}\left(\hat{Y_{ut}} - \log_2\,(2Y_{ut})\right)^2}{|E|} \tag{2}$$

While it is expected that users and already listened tracks will show a high strength, distance aims at weighting the actual loss of the network in a way that interactions between users and tracks that are farther away (i.e., do not belong to the user filter bubble) will carry a higher weight that interactions in the same bubble. Then, this loss definition favours recommendation diversity by learning the structure of filter bubbles without explicitly finding them, allowing for more flexibility in bubble definition. In this sense, hyperparameter $\beta$ allows tuning weight distances during training, i.e., whether to receive closer and more accurate recommendations ($\beta$ close to zero, which reduces the relevance of distance and better learning user preferences in their closer communities ) or farther and more diverse recommendations ($\beta$ close to one). After preliminary evaluations, $\beta$ was set to 0.75.

In addition to scrobbles, which represent the positive interactions of a user with the listened tracks, we applied a negative sampling with a $1:1$ ratio, where $Y_{ut}$ was set to 0.5 so that $\log_2\,(2Y_{ut})$ would be zero. Unlike the BPR [24] framework, the network independently processes negative and positive samples, learning that negative instances should decrease the strength of the interaction. $d(u,t)$ was replaced by $d_{neg}\,(u,t) = 1 - d\,(u,t)$ to prevent the negative sampling from penalizing tracks that are unlikely to have already been listened but could be relevant.

A 10% edge dropout was introduced to the GCN training to avoid overfitting over the graph structure and improve generalization. This is similar to DropEdge [25], with the difference that dropout is applied over $\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}$ instead of $\hat{A}$, thus avoiding recomputing $\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}$ for each mini-batch, thus increasing the efficiency.

***Precomputed distances.*** Distance $d\,(u,t)$ in the loss function was precomputed based on user and track embeddings that are different from those of the model to avoid introducing a dependency between the model and the distance function. Training is based on Word2Vec [19], where the goal is to determine whether a user listened to a track, regardless of its scrobbles. A binary cross-entropy loss function (Eq. 3) is used, where $w_u$ and $\hat{w}_t$ represent the user and track embeddings, $\sigma$ is the sigmoid function, and $A_{ut}$ is set to 1 if user $u$ listened to track $t$, and 0 otherwise. Negative sampling is also applied to this training process with a ratio $1:1$, where a user and a track are randomly selected.

$$L_{dis} = \frac{-\sum\left(A_{ut}\log\left(\sigma\left(w_u \cdot \hat{w}_t\right)\right) + (1 - A_{ut})\log\left(1 - \sigma\left(w_u \cdot \hat{w}_t\right)\right)\right)}{|N|} \tag{3}$$

Distance $d$ is defined by the cosine between user and track embeddings (Eq. 4 and Eq. 5), where $\mu_{\cos}$ and $\sigma_{\cos}$ are the mean and standard deviation of $\cos(w_u, w_t)$ for each user-track pair. This definition aims at better learning the relation between farther users and tracks (for which there are fewer and more diverse examples) rather than closer tracks and users. This formulation centers on the media of the cosine between users and tracks during training, adjusting the values according to their deviation, providing a better adaptation than the traditional cosine distance to situations in which user and track vectors are close. Clipping values were set to reduce the influence of users over the loss function.

$$d\,(u,t) = \begin{cases} 0.1 & \text{if } d_{raw}\,(u,t) < 0.1 \\ 0.9 & \text{if } d_{raw}\,(u,t) > 0.9 \\ d_{raw}\,(u,t) & \text{otherwise} \end{cases} \tag{4}$$

$$d_{raw}\,(u_i, u_j) = 1 - \frac{\cos\left(w_u, \hat{w}_t\right) - \left(\mu_{\cos} - 2\sigma_{\cos}\right)}{4\sigma_{\cos}} \tag{5}$$

## 4 EXPERIMENTAL SETTINGS

***Data collection.*** Evaluation was based on data collected from *Last.fm*. We focused on the track listening history and the users' social networks. Based on the social interactions collected by Zhang

| **#Users** 3,307 | **#Tracks** 252,014 | **#Artists** 28,540 |
|---|---|---|
| | avg (± std) | 25% - 50% - 75% quantiles |
| Tracks per User | 912 (± 1266) | 624 - 854 - 1043 |
| Scrobbles per User | 48 (± 73) | 11 - 28 - 58 |
| Social relations per User | 86 (± 85) | 29 - 59 - 118 |

**Table 1: Data collection details**

et al. [32][3], we selected the top 5% of users with the highest number of social connections and scrobbles. We collected the scrobble history for each of the 3, 307 selected users using the Last.fm API. From the set of over 1 million tracks listened by the selected users, we selected approximately $252k$ tracks with the highest number of listeners among the selected users, with 99% of users associated with over 40 songs. For each selected track, we collected the total number of scrobbles and listeners, tags, artist (and their tags) and Spotify audio features[4]. To make features' scores comparable, tempo was standardized, and loudness was applied a logarithm transformation. Table 1 summarizes the basic statistics of the collected data[5].

***Baselines.*** `MRecuri` was compared to 9 different recommenders. When available, the original implementations were used. Parameters were optimized as described in the original studies. First, two trivial and non-personalized reference baselines: **popularity** (i.e., tracks with the highest ratio of scrobbles and listeners were recommended), and a **random** recommender, as a lower bound reference. Second, a **content-based** technique, in which, based on preliminary evaluations, tracks were represented by the average Word2Vec embedding of their tags, and users were represented by the centroid of the listened track vector. Then, recommendations were made based on the cosine similarity between a track and the user representation. Third, several traditional and state-of-the-art user-item recommender techniques: i) **ImplicitMF** [13], a top-performing matrix factorization technique based on a factor model tailored for implicit feedback settings. ii) **GraphRec** [8], a graph neural network that jointly represents social interactions, item features, and ratings. iii) **MultVAE** [15], a neural generative model based on variational autoencoders and multinomial conditional likelihood. Finally, we considered diversity-oriented techniques: i) **Rank Aggregation** (RA) techniques based on Copeland's ranking method. The base rankings were built using ImplicitMF and listing items according to their diversity and novelty scores (as defined in Section 4) over their content-based representation. ii) An **MMR** [3] inspired technique. The base recommender was set to ImplicitMF. Diversity was measured considering the cosine similarity of the content-based representation of items, as in [6]. iii) **VC** Vargas and Castells [31]. Profiles were based on the top-20 tags, and base recommendations were obtained based on ImplicitMF.

***Evaluation metrics.*** In addition to relevance-based metrics (precision, nDCG), recommendations were assessed considering how they help users outing from their bubbles [27]. We considered variations of intra-list dissimilarities [26] to assess *diversity* (i.e., differences within an experience, in this case the set recommended items) and *novelty* (i.e., differences between past and present experiences, in this case, the known and recommended items) [5]. Following Nguyen et al. [21], we consider diversity and novelty as proxies for measuring the filter effect. Then, a diversity/novelty increment is assumed to imply a decrease in the filter bubble effect.

Dissimilarities were measured based on the Euclidean distance of structural and content-based vectors. For computing the structural distances, each user was represented by a vector defining the consumption rate of tracks belonging to the different track communities [10]. Such communities were discovered based on a co-listened track graph (i.e., two tracks are connected if they were listened by the same user) using the Louvain algorithm [2]. On the other hand, content-based dissimilarity was computed based on the Word2Vec representation of tracks' tags[6]. User representation was defined as the average representation of the listened tracks.

To compute diversity and novelty, we assumed that users would listen to all recommendations, i.e., all recommendations were deemed correct. Although this assumption might not be realistic [27], it allows observing how recommendations could shape the future network. Diversity was computed by comparing all recommended tracks with each other, while novelty was computed by comparing all listened tracks (training set) vs. all recommended tracks.

***Implementation details.*** The model was implemented on TensorFlow. The optimizer was set to Adam with a learning rate of $1e-3$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The dimension of all embeddings was set to 64, while for the dense layers, it was set to 32. The learning process was stopped once no loss changes were observed, reaching convergence after 10 epochs[7]. The model was trained on an Asus Predator Helios 300 with 16Gb RAM, an i7-11800H, and an NVidia GeForce RTX 3060 6Gb. Training and recommendations took approximately 6 hs and 15 minutes, respectively.

Evaluation was performed in an offline setting over the same data partitions. User scrobbles were temporally split. For each user, the first 70% of listened tracks were used as training, while the remaining tracks were used as the test set. We defined a threshold to select the top-$K$ recommended tracks, where $K$ was set to 10 as over 96% of users had 10 tracks in the test set. Recommendations were considered correct if they appeared in the test set.

# 5 EVALUATION

Table 2 presents the obtained results. We report the average score across all users and the standard deviation. We also report the results of a paired statistical analysis with an alpha value of 0.01. For each metric, the best three results are shown in **<u>bold+underline</u>**, **bold** and <u>underline</u>, respectively. For reporting the diversity and novelty of the original user-track graph, we computed how diverse/novel are tracks in the test set regarding the tracks in the training set.

Random and GraphRec achieved the lowest relevant results (with differences up to 71% and 85% in terms of precision and nDCG). Conversely, they achieved high diversity/novelty results, with GraphRec among the best performing techniques. There is a trade-off between

[3] The original set of users can be retrieved from https://www.aminer.cn/cosnet.
[4] https://developer.spotify.com/documentation/web-api/reference/#object-audiofeaturesobject
[5] The final retrieved set of users and tracks with their metadata and the resulting graphs are available at the companion repository: https://github.com/tommantonela/umap2022-mrecuri.
[6] As an example, according to their tags, "Yesterday" (The Beatles) would be similar to "Love of my life" (Queen), but dissimilar to "Sunset Garage" (Duran Duran).
[7] More details and implementations can be found at the companion repository.

| | Precision | nDCG | Content-based dissimilarities | | Structural dissimilarities | |
|---|---|---|---|---|---|---|
| | | | Diversity | Novelty | Diversity | Novelty |
| MRecuri | 0.28 ± 0.198 | 0.648 ± 0.227 | **0.471** ± 0.014 | **0.547** ± 0.031 | 0.128 ± 0.009 | **0.387** ± 0.092 |
| Random | 0.101$^*$ ± 0.004 | 0.446$^*$ ± 0.201 | **0.496** ± 0.028 | **0.485** ± 0.023 | **0.224** ± 0.075 | 0.282$^*$ ± 0.045 |
| Popularity | 0.203 ± 0.133 | 0.581 ± 0.237 | 0.402$^*$ ± 0.017 | 0.46$^*$ ± 0.026 | 0.131 ± 0.015 | 0.279$^*$ ± 0.031 |
| Content-based | 0.168$^*$ ± 0.124 | 0.522 ± 0.215 | 0.173$^*$ ± 0.062 | 0.388$^*$ ± 0.046 | 0.219 ± 0.061 | 0.247$^*$ ± 0.033 |
| ImplicitMF | **0.366** ± 0.238 | **0.699** ± 0.226 | 0.376$^*$ ± 0.061 | 0.417$^*$ ± 0.042 | 0.146 ± 0.049 | 0.228$^*$ ± 0.036 |
| GraphRec | 0.1$^*$ ± 0.03 | 0.391$^*$ ± 0.149 | **0.497** ± 0.011 | **0.542** ± 0.012 | **0.325** ± 0.007 | **0.317**$^*$ ± 0.022 |
| MultVAE | **0.348** ± 0.245 | **0.686** ± 0.231 | 0.414$^*$ ± 0.07 | 0.457$^*$ ± 0.049 | 0.176 ± 0.061 | 0.236$^*$ ± 0.039 |
| Rank Aggregation | 0.241 ± 0.164 | 0.57 ± 0.214 | 0.453$^*$ ± 0.118 | 0.531 ± 0.043 | 0.179 ± 0.054 | 0.238$^*$ ± 0.038 |
| MMR-inspired | 0.247 ± 0.172 | **0.687** ± 0.245 | 0.41$^*$ ± 0.152 | 0.515 ± 0.04 | 0.176 ± 0.054 | 0.236$^*$ ± 0.037 |
| VC | 0.288 ± 0.189 | 0.6 ± 0.216 | 0.421$^*$ ± 0.058 | 0.469$^*$ ± 0.041 | 0.195 ± 0.055 | 0.243$^*$ ± 0.037 |
| Original user-track graph | - | - | **0.472** ± 0.066 | 0.484 ± 0.047 | 0.229 ± 0.045 | 0.246 ± 0.031 |

**Table 2: Recommendation results comparison for $k = 10$. (* indicates statistically significant differences favouring MRecuri)**

relevance and diversity/novelty, as techniques achieving high relevance also achieved low diversity/novelty. Popularity was an exception as it achieved a better balance between relevance and diversity, despite showing a lower diversity/novelty than the original graph. These results might indicate that despite being popular tracks, they might be different from what users usually listen to, thus improving diversity/novelty. On the other hand, ImplicitMF and MultVAE achieved the highest precision and nDCG while achieving lower diversity/novelty than the original graph.

Regarding the diversity-oriented techniques, the highest relevance was observed for MMR, while the most diverse and novel results were observed for RA. Except for structural diversity, results were outperformed by MRecuri. The RA baseline's best results were achieved using Copeland's method to aggregate three rankings: ImplicitMF and diversity and novelty rankings based on a track representation including both tag embeddings and Spotify features. On the other hand, the best results for the MMR baseline were observed when representing tracks only using the tag embeddings and setting the diversity weight to 0.9. For VC, only minor differences were observed when varying $\lambda$.

MRecuri was among the best performing techniques for most metrics (even achieving the highest novelty), including precision and nDCG, with average improvements of 56% and 19%, respectively. In terms of diversity/novelty, except for structural diversity, MRecuri was able to improve the scores of the original graph. The average diversity/novelty differences favouring MRecuri regarding the simpler, state-of-the-art, and diversity-oriented baselines were 25%, 9%, and 15%, respectively. These results show that diversity/novelty can be improved while providing relevant recommendations. In general, novelty was higher than diversity, meaning that even when recommending similar tracks, they differed from those in the listening history. The lowest results for MRecuri were observed for structural diversity, implying that the recommended ranking included tracks with similar co-listened patterns. Nonetheless, MRecuri also achieved the highest structural novelty results, implying that recommendations were outside the influence of the co-listened community of the already listened tracks, which can effectively broaden users' music perspectives.

# 6 CONCLUSIONS

This work presented MRecuri, a music recommender fostering content diversification in a filter bubble setting. MRecuri focused on implicitly characterizing filter bubbles based on user listening history, social interactions, and a music knowledge graph. The performed offline evaluation showed the potential of MRecuri for expanding users' listening diversity and novelty compared with state-of-the-art techniques, while maintaining competitive precision and nDCG.

Several aspects could be tackled in future works. First, perform a more extensive evaluation in large-scale scenarios to fully assess the technique's usefulness, generalizability, and scalability. Second, perform an ablation study to assess the contribution (or effects) of the different components, particularly, the interplay of information sources and their contributions to the final recommendations, and the effect of negative sampling and the defined distance metric. Third, considering the particularities of music consumption, in which tracks are rarely isolatedly consumed, MRecuri could be extended to include information of the listening history as an ordered sequence. Fourth, the technique could be enriched to provide explanations and thus increase the transparency and trust of recommenders. Finally, a user study should be conducted to assess recommendations' perceived relevance and diversity.

# REFERENCES


[1] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. 2020. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of The Web Conference 2020*. 2155–2165.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[3] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[4] Giovanna Castellano, Giovanni Sansaro, and Gennaro Vessio. 2021. ArtGraph: Towards an Artistic Knowledge Graph. *arXiv preprint arXiv:2105.15028* (2021).

[5] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. *Novelty and Diversity in Recommender Systems*. Springer US, Boston, MA, 881–918. https://doi.org/10.1007/978-1-4899-7637-6_26

[6] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. 2014. An analysis of users' propensity toward diversity in recommendations. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 285–288.

[7] Tomislav Duricic, Dominik Kowald, Markus Schedl, and Elisabeth Lex. 2021. My friends also prefer diverse music: homophily and link prediction with user preferences for mainstream, novelty, and diversity in music. *arXiv preprint arXiv:2111.00562* (2021).

[8] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. ACM, New York, NY, USA, 417–426. https://doi.org/10.1145/3308558.3313488

[9] Sarah Frost, Manu Mathew Thomas, and Angus G Forbes. 2019. Art i don't like: An anti-recommender system for visual art. In *Proceedings of Museums and the Web*.

[10] Quentin Grossetti, Cédric du Mouza, and Nicolas Travers. 2019. Community-Based Recommendations on Twitter: Avoiding the Filter Bubble. In *Web Information Systems Engineering – WISE 2019*, Reynold Cheng, Nikos Mamoulis, Yizhou Sun, and Xin Huang (Eds.). Springer International Publishing, Cham, 212–227.

[11] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and sequential user embeddings for large-scale music recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 53–62.

[12] Christian Hansen, Rishabh Mehrotra, Casper Hansen, Brian Brost, Lucas Maystre, and Mounia Lalmas. 2021. Shifting consumption towards diverse content on music streaming platforms. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 238–246.

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 263–272.

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[15] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[16] Omar Lizardo. 2006. How cultural tastes shape personal networks. *American sociological review* 71, 5 (2006), 778–807.

[17] Feng Lu and Nava Tintarev. 2018. A Diversity Adjusting Strategy with Personality for Music Recommendation.. In *IntRS@ RecSys*. 7–14.

[18] Madison Grace McClung and Angèle Christin. 2018. Filter Bubbles And Music Streaming : The Influence of Personalization And Recommendation Algorithms on Music Discovery Via Streaming Platforms.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc.

[20] Jeremy Wade Morris and Devon Powers. 2015. Control, curation and musical experience in streaming music services. *Creative Industries Journal* 8, 2 (2015), 106–122.

[21] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of the 23rd International Conference on World Wide Web* (Seoul, Korea) *(WWW '14)*. ACM, New York, NY, USA, 677–686. https://doi.org/10.1145/2566486.2568012

[22] Róbert Pálovics and András A Benczúr. 2015. Temporal influence over the Last. fm social network. *Social Network Analysis and Mining* 5, 1 (2015), 4.

[23] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think.* Penguin.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[25] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903* (2019).

[26] Dimitris Sacharidis. 2019. Diversity and Novelty in Social-Based Collaborative Filtering. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization* (Larnaca, Cyprus) *(UMAP '19)*. ACM, New York, NY, USA, 139–143. https://doi.org/10.1145/3320435.3320479

[27] Javier Sanz-Cruzado and Pablo Castells. 2018. Enhancing Structural Diversity in Social Networks by Recommending Weak Ties. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. ACM, New York, NY, USA, 233–241. https://doi.org/10.1145/3240323.3240371

[28] Markus Schedl, Peter Knees, Brian McFee, Dmitry Bogdanov, and Marius Kaminskas. 2015. Music recommender systems. In *Recommender systems handbook*. Springer, 453–492.

[29] Maria Taramigkou, Efthimios Bothos, Konstantinos Christidis, Dimitris Apostolou, and Gregoris Mentzas. 2013. Escape the bubble: Guided exploration of music preferences for serendipity and novelty. In *Proceedings of the 7th ACM conference on Recommender systems*. 335–338.

[30] Antonela Tommasel, Juan Manuel Rodriguez, and Daniela Godoy. 2021. I Want to Break Free! Recommending Friends from Outside the Echo Chamber. In *Fifteenth ACM Conference on Recommender Systems* (Amsterdam, Netherlands) *(RecSys '21)*. Association for Computing Machinery, New York, NY, USA, 23–33. https://doi.org/10.1145/3460231.3474270

[31] Saúl Vargas and Pablo Castells. 2013. Exploiting the diversity of user preferences for recommendation. In *Proceedings of the 10th conference on open research areas in information retrieval*. Citeseer, 129–136.

[32] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S Yu. 2015. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1485–1494.