Supporting Serendipitous Recommendations With Knowledge Graphs

Oliver Baumann[1], Mirco Schoenfeld[1]

[1]University of Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany

Abstract

Recommender systems are commonly designed and evaluated with high precision and accuracy in mind. Optimising systems for these metrics alone can, however, lead to a decrease in overall collection coverage of recommended items, and over-emphasize popular content. In order to present useful suggestions to users, it has been argued that a recommender system should also provide novel and diverse items different to what the user has experienced in the past. This closely ties in with the notion of serendipity, i.e., making a surprise discovery akin to a "happy accident", that is nevertheless interesting and relevant. We implement a recommender system based on a knowledge graph of musical items with serendipity, novelty and diversity in mind. Using acoustic features as contextual information for vertices in the graph, we explicitly select content dissimilar from the user's previous experience. We compare our results to a set of baseline algorithms and find that the investigated approach is able to recommend diverse and novel items.

Keywords

knowledge graphs, recommender systems, serendipity, novelty, diversity, attributed graphs

1. Introduction

Finding relevant content in potentially vast repositories of information is a key challenge for users. Whether it's finding interesting products in an online shop, valuable social contacts in an Online Social Network, or items in a digital library, the domain is often so large in size and broad in variety that gaining a complete overview is at best hard, at worst impossible. Recommendation systems can aid in the process of discovery by suggesting items the user has not been exposed to previously, but that may be of interest to them. A common assessment of recommender systems is by *precision* and *recall*, as well as *top-n* variants thereof that only take into account the *n* highest ranked items. While precision is a measure for how relevant the suggested items are to the user, recall measures the extent to which all relevant items are retrieved. A system exhibiting high precision and recall is hence able to accurately predict the user's interests, and returns all relevant items.

It has been argued [1, 2, 3], however, that optimising a recommender system for precision and recall alone disregards a variety of other dimensions that prove useful both in assessing and

CIRCLE 2022: Joint Conference of the Information Retrieval Communities in Europe, July 04–07, 2022, Samatan, Gers, France

oliver.baumann@uni-bayreuth.de (O. Baumann); mirco.schoenfeld@uni-bayreuth.de (M. Schoenfeld)
https://baumanno.de (O. Baumann); https://mircoschoenfeld.de (M. Schoenfeld)
0000-0003-4919-9033 (O. Baumann); 0000-0002-2843-3137 (M. Schoenfeld)

designing recommender systems. Notions such as novelty, diversity, and serendipity can benefit users when factored into the design of such systems by enabling them to venture outside the space of familiar content and discover new items they have not been exposed to.

A serendipitous discovery can be compared to a "happy accident": a finding that is unexpected and novel, but nevertheless relevant. For instance, Iaquinta et al. [4] argue that novelty is a key contributor to serendipitous discovery, as an item the user has not experienced is more likely to result in a "surprise finding" than one they already know, but disregarded as irrelevant or uninteresting. As such, it is hard to measure serendipity directly, and thus previous work [5, 1, 2, 6] commonly considers notions of serendipitous discovery alongside diversity and novelty.

In this work, we explore a graph-based approach to generating novel and diverse recommendations, thus contributing to the body of work concerned with *beyond accuracy* measures and serendipity in recommender systems. We abstract parts of the musical domain into a knowledge graph, consisting of tracks and their genre-annotations. Based on this graph, relevant, yet novel and diverse candidate tracks for recommendation are determined and ranked with scoring functions that intentionally discount items similar to the user's past experience. A set of content- and graph-based measures taken from existing literature are employed to evaluate the results in terms of novelty and diversity. We believe that our approach can be generalized to fit other domains than music, providing the items subjected to recommendation can be compared in terms of similarity or distance, and a simple ontology can be utilized to construct the required graph.

The remainder of this work is structured as follows: in Section 2, we provide an overview of previous work; Section 3 outlines the datasets used in our analysis; in Section 4 we present our method for finding diverse and novel recommendations; Section 5 presents our main findings and compares them to a set of baselines; and Section 6 concludes this work.

2. Related Work

Graph-based approaches for recommender systems have been used to model relationships between users, between items, and between both in conjunction, e.g., in the context of digital libraries [7], Online Social Networks [8], or music [9]. Apart from items or users, knowledge graphs provide the possibility to model entities as part of an ontology, along with their relations [10, 11].

The knowledge graph employed in the paper at hand is an instance of an attributed graph, as vertices carry additional context-information. Recently, Schoenfeld and Pfeffer [12] proposed an approach for calculating shortest path-based centrality-measures on attributed graphs. At each step during path discovery between two nodes, a distance function is evaluated, its input being vertex-attributes or attribute-vectors. Paths exceeding a provided threshold can then be pruned from the search tree, enabling vertex-specific views of a graph and restricting graph traversal to only a subset of vertices and incident edges that meet a custom decision criterion.

Acoustic features of songs have been previously used to, e.g., analyze the quality of music recommendations for "beyond mainstream" users [13]. Users are assigned a mainstreaminess score based on a user's artist-playcount and the global artist-playcounts. By employing a

two dimensional embedding of tracks' acoustic features, four clusters of musical genres are detected. Users are then assigned to clusters based on the number of tracks in the cluster they have listened to. A set of baseline recommendation algorithms is evaluated against these user-clusters, showing statistically significant results across groups, thus highlighting the importance of user-modelling.

Zangerle et al. [14] have also used acoustic features to create culturally-aware user models. Users' listening preferences are modelled based on the tracks they have listened to by aggregating the tracks' acoustic feature vectors. The model is further enriched with Hofstede's six cultural dimensions based on users' country of origin. Using XGBoost, precision and recall are evaluated for the music-cultural and only-music models, with the combined music-cultural model outperforming the baselines, and the music-model performing second best. These findings indicate that modelling users through acoustic feature vectors has a positive impact on recommendation accuracy.

Ge et al. [3] argue that metrics applied to evaluation of recommender systems should incorporate a notion of quality of recommendations rather than merely testing for accuracy. They present a set of measures for coverage and serendipity and approach the latter through the notion of unexpectedness of results as compared to a primitive prediction model. The presented measure for serendipity constrains the unexpected items to also be useful, an evaluation that ultimately only a real user of the system can make. Concluding, they argue that in order to foster serendipity, a system should aim for high catalog coverage, so as to include rarely or low rated items.

3. Datasets

We base our analysis on three datasets: the LFM-1b dataset [15], the CultMRS dataset [14], and a subset of LFM-1b annotated with musical genres[1].

The LFM-1b dataset contains more than one billion listening events across 120 000 users and 32 million tracks. The data was collected between January 2013 and August 2014 by querying listening histories of users on the music-streaming platform *last.fm* through the platform's public Web-API. For the purposes our analysis, a listening event is a tuple (u, t) consisting of a unique, anonymous user-ID u and a unique track-ID t; additional identifiers for artists and albums are available, but not relevant to this study. Track-IDs can be unambiguously related to a single musical item, and tuples need not be unique, i.e., multiple events for the same user and track are valid. Please note that, while the timestamp when the user began listening to a track is recorded in the dataset, no indication can be made as to whether they listened to it completely, or skipped to a new item at some point through the track.

The EmoMTB dataset contains genre-annotations for a subset of 533 970 tracks in the LFM-1b collection. Genres were fetched by querying the *last.fm*-API for the user-generated tags on a track and matched against two dictionaries of musical genres. In this dataset, a track is annotated with an average of 4.92 ($\pm$4.29) genres, the median is 4.

A further extension to the LFM-1b collection is the CultMRS dataset that enriches 3 471 884 tracks with acoustic features. The data was collected for 55 190 users providing country-

[1]This dataset was kindly provided to us by the authors of the EmoMTB-project [16].

information on their *last.fm* profile. Acoustic features were obtained through the public *Spotify* Web-API[2]. We defer a detailed definition of these features to Zangerle et al. [14] and the API documentation published by *Spotify*. Out of the set of available features, we use *acousticness*, *danceability*, *energy*, *instrumentalness*, *liveness*, *speechiness*, *tempo*, and *valence* and omit *key*, *loudness*, and *mode* as we don't believe these to be indicative of users' listening behaviour. All feature values range in $[0, 1]$, with the exception of tempo, which denotes the beats-per-minute (BPM) as a floating-point number. Using linear min-max normalization, we normalize tempo into the $[0, 1]$ range, following the approach of Zangerle et al. [14].

As Kowald et al. [13] noted, there is a tendency for *last.fm* users to assign coarse-grained genres such as "pop" and "rock" to tracks. Since these overly prevalent genres would distort our model of the musical domain, we follow the approach from Kowald et al. [13] and use inverse document frequency (IDF) to remove overly popular genres from the EmoMTB-dataset. To this end, we view genres as terms and tracks as documents and determine a genre g's IDF score as:

$$IDF(g) = \log \frac{\#tracksInDataset}{\#tracksAnnotatedWith\_g}$$

By visual inspection of the IDF scores of the top 100 genres we choose a cutoff score of 0.903 (dashed line in Figure 1), and omit the genres "rock", "pop", "metal", and "alternativerock" (in ascending order of their respective IDF score) from our analysis. This removes 7 227 tracks from the dataset (ca. 2.80%) that have no other except the excluded genres assigned. All remaining tracks have at least one genre, with a mean of 4.59 ($\pm$3.98) and a median of 3.

For further analysis, we select all tracks that appear both in the filtered EmoMTB- and CultMRS-dataset, which results in a total of 250 654 tracks and 2 126 genres.

Prior to sampling the set of users for which we determine recommendations, we remove users appearing as outliers in terms of the number of distinct tracks they listened to. Using the full LFM-1b set of listening events, we count the unique tracks each user has listened to and perform outlier detection using median absolute deviation (MAD) [17] with a "very conservative" threshold of 3, i.e., we consider a user an outlier if the number of tracks they have listened to falls outside of three median absolute deviations from the median. From the resulting distribution, we draw a random sample of 1 000 users that have listened to between 200 and 500 tracks. From this sample, we select all tracks that also appear in our annotated sets, resulting in a collection of 1 000 users that listened to 106 679 distinct tracks across 1 898 genres.

4. Methods

In this section, we report on the methods employed to construct a knowledge graph of the musical domain at hand. We then outline how we find candidate tracks for recommendation and subsequently score these to arrive at a ranked list of tracks we recommend for a given user.

[2] https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

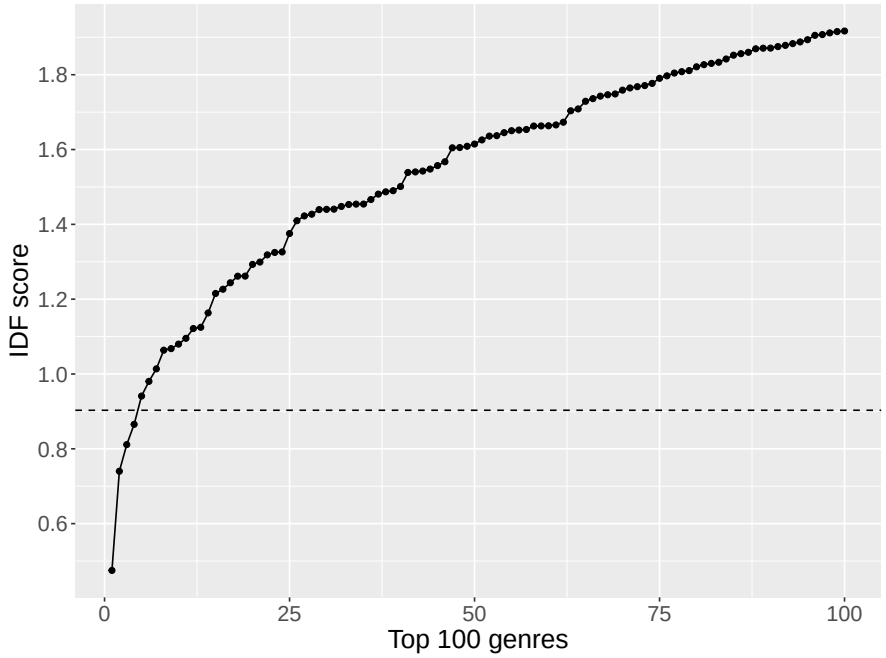

Figure 1: IDF scores of top 100 genres in ascending order; the dashed line indicates the cutoff at 0.903

4.1. Knowledge Graph

Tracks and genres are modelled as an undirected, bipartite graph $G = (U, V, E)$ consisting of two types of vertices: tracks U and genres V. For two vertices of distinct types the edge $(u, v) \in E$ if the track-genre-annotation is present in the dataset.

We use the merged CultMRS and EmoMTB datasets (see Section 3) to construct this graph, which has 252 780 vertices and 1 149 345 edges. The mean degree is 9.09 ($\pm$201.43), the median is 3. As can be seen in Figure 2, the degree distribution for the entire graph is long-tailed, with the majority of vertices exhibiting a degree of 500 or less. Across all track-vertices, the mean degree is 4.59 ($\pm$3.98) with a median of 3, and across genre-vertices the mean is 540.61 ($\pm$2130.59) with a median of 22.

For each user, we extract a bipartite subgraph $g = (U_u, V_u, E)$ from G containing only tracks U_u the user has listened to, along with their genres V_u. Furthermore, for each subgraph we determine the one-mode projection onto genre-vertices, $g_{Genre} = (V_u, F)$, where F denotes the set of edges and an edge $(v_1, v_2) \in F$ if these two genre-vertices share a common track-vertex as their neighbour in the user-subgraph g.

4.2. Generating Recommendations

To find candidate tracks for recommendation, we attempt to close triangles in users' genre-subgraphs g_{Genre}. This relates to the idea of reciprocity in social networks, where closed triangles indicate mutual ties among people knowing each other ("a friend of a friend is also

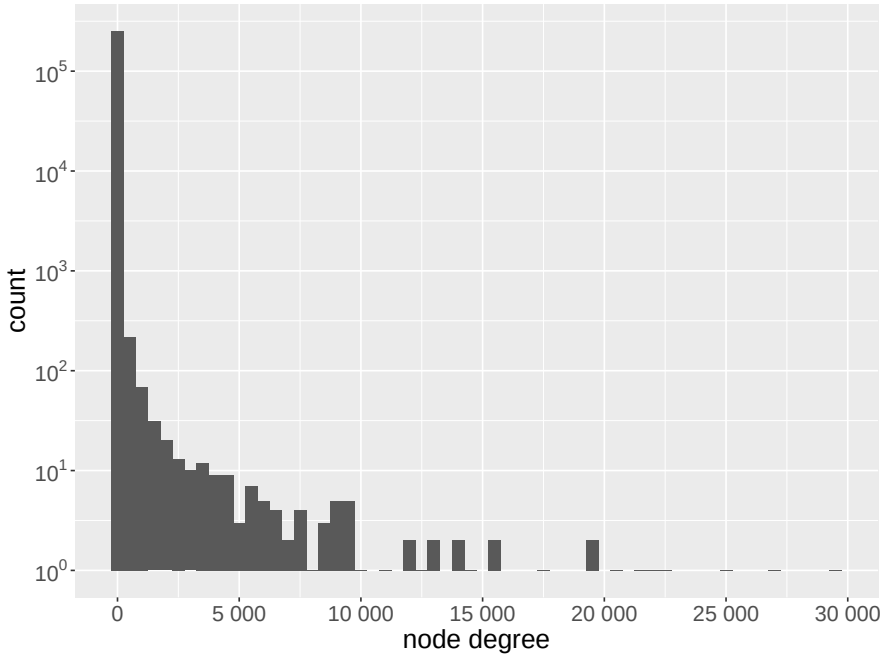

Figure 2: Degree distribution in bipartite knowledge graph (bin-width: 500, y-axis log-scaled).

my friend")[18]. Transferring this idea to the genre-subgraph, two genre-vertices may have a mutual neighbour the user is unaware of. Incorporating this genre into the user's listening experience has the potential of venturing deeper into the global genre-graph, while maintaining a strong connection with existing preferences and thus constituting a relevant discovery. As we wish to form connections among genres in the one-mode projection, we need to introduce vertices into the bipartite graph that are of the track-type in order for new edges to manifest in the projection.

Thus, we determine, for all pairs of genre-vertices $v_i, v_j \in g$ the user has listened to, the set-intersection of their (open) neighborhood $N_G(v)$ in the main bipartite graph G, resulting in in a set of candidate tracks C that are annotated with both genres:

$$C = \bigcup_{v_i, v_j \in g} N_G(v_i) \cap N_G(v_j)$$

From this set, we exclude any track-vertices u that are already present in the user's listening history:

$$C' = C - \{u : u \in g\}$$

The remaining candidate tracks in C' are scored to obtain a final ranked list of recommendations.

4.3. Scoring

We determine a ranked list of tracks by applying a scoring function to each candidate track. In this work, we evaluate two scoring functions parameterized on a similarity metric and overall popularity in order to balance similar items against popular ones.

We use cosine similarity on vectors of acoustic features to determine how close two musical items are perceptually. Each track is assigned a feature-vector $\vec{a}$ based on the CultMRS dataset. For each user, we compute a profile-vector $\vec{p}$ resembling their listening profile by summarising all feature-vectors of tracks in their history. Following Zangerle et al. [14], we apply MAD to remove outliers from users' listening histories in terms of acoustic features. We consider a track an outlier if any component of its feature-vector falls outside of three MADs around the median of that feature. The remaining tracks' vectors are summarised into a single profile-vector using the median of each component.

As an indicator of popularity, we count in how many listening histories within our sample a track appears, and scale this value into the range $[1, 1000]$ using min-max normalization. In our scoring functions, we opt for the square-root of this value to alleviate the impact of large popularity scores.

Both scoring functions are designed to assign higher scores to tracks with low popularity and thus favour items that appear in the global long tail. In the same way as this increases the recommender system's coverage of the entire collection of music, it also opens up the door for recommendations to feature items that appear novel and unexpected to the user. Here, we follow Ge et al. [3] who argue that serendipitous discovery may happen where relevant, yet unexpected findings are encountered.

4.3.1. Popularity-discounted similarity

In order to assign lower scores to highly popular tracks, we discount the cosine similarity of a user's profile vector $\vec{p}$ and a track's feature vector $\vec{a}$ by the track's popularity $pop(t)$:

$$r(t) = \frac{cossim(\vec{p}, \vec{a})}{\sqrt{pop(t)}}$$

This scoring is applied to all tracks in the candidate set, and the resulting scores are put in decreasing order. The top 20 tracks are then subjected to evaluation.

4.3.2. Logistic scoring

To discount globally popular tracks while boosting low similarities and hence potentially novel tracks, we employ scoring based on the logistic function:

$$r(t) = 1 - \frac{1}{1 + e^{-cossim(\vec{p},\vec{a})\sqrt{pop(t)}}}$$

This is, in essence, a logistic function $f(x) = \frac{1}{1+e^{-k(x-x_0)}}$ with $k = cossim(\vec{p}, \vec{a})$, $x = \sqrt{pop(t)}$ and $x_0 = 0$; we use the complement to ensure a declining function rather than an increasing one.

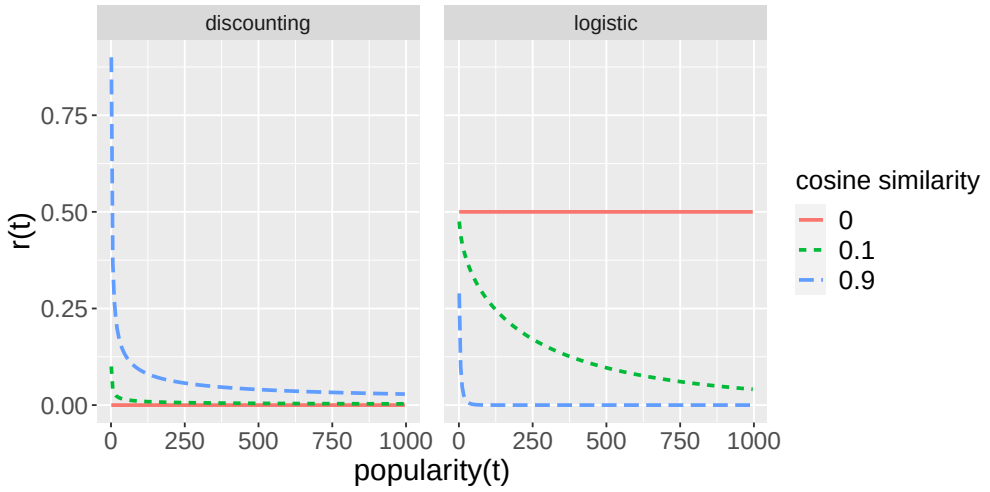

Figure 3: Scoring functions (left: popularity-discounted scoring; right: logistic scoring)

Figure 3 shows the two scoring functions, evaluated for three example similarity values across the full popularity range. Note that for logistic scoring, two maximally dissimilar vectors would be assigned a score of 0.5, regardless of popularity.

5. Evaluation and Discussion

For each of the 1 000 users in our dataset, we evaluate how diverse and how novel the top-20 recommendations are using a set of measures outlined in the following sections. In addition, we compare our results against a set of baseline algorithms, which we implement using the Python-library Surprise [19]. As the algorithms we implement require explicit ratings, whereas our dataset contains implicit ratings in the form of listening events, we count how often a track appears in a user's listening history and scale this value into the range $[1, 1000]$ using min-max normalization.

The baselines that we use from Surprise are: *NormalPredictor*, which samples a user-item-rating from a normal distribution with an estimated mean and standard deviation from the existing user-item-ratings; *BaselineOnly*, which predicts a rating using a baseline estimate of user-average and item-average deviations from the global mean rating [20]; and *NMF*, non-negative matrix factorization [21].

All baseline algorithms are trained on the entire dataset, and predictions are generated for the "anti-training-set", i.e., all user-item pairs missing from the training set. NMF is trained with the default parameters set by the Surprise library. As with our own approach, we return the 20 highest-ranked predictions.

In the following evaluation, we report measures for diversity and novelty across our two scoring functions (see Section 4.3) "popularity-discounted similarity", which we shall refer to as *PDS*, and "logistic scoring", which we refer to simply as *logistic*. The same measures are computed for the baseline recommendations and contrasted with our approaches.

Table 1
Evaluating Intra-list Diversity

algorithm	mean (±SD)
PDS	0.0043 (±0.0039)
logistic	0.0505 (±0.0545)
BaselineOnly	0.2178 (±0.0024)
NMF	0.2300 (±0.0400
NormalPredictor	**0.2342** (±0.0398)

5.1. Diversity

Diversity of recommendations tries to capture how different recommendations are, either among themselves, i.e., within a recommendation list, or on a global level, e.g., which proportion of all items in the catalog appear as the output of a recommender system [6].

We employ two measures concerned with recommendation lists: *Intra-list Diversity* and *Structural Diversity*, a measure tailored to graphs.

5.1.1. Intra-list Diversity

Simply put, Intra-list Diversity (ILD) measures the average pairwise distance among all items in a recommendation list R with respect to a distance measure d:

$$ILD(R) = \frac{1}{|R|(|R|-1)} \sum_{i \in R} \sum_{j \in R} d(i,j)$$

In our case, $|R| = 20$, and d is the cosine distance, i.e., the complement to cosine similarity, of the recommended tracks' acoustic feature vectors.

As Table 1 shows, NormalPredictor returns the overall most diverse recommendation lists. All baseline algorithms perform considerably better in terms of ILD than our approach. Of our approaches, logistic scoring performs better than PDS, indicating that by scoring dissimilar content higher on a per-item basis, more diverse results can be achieved. To assess whether the differences between our scoring functions are statistically significant, we conduct a Wilcoxon signed-rank test as the data are not normally distributed (Shapiro-Wilk test, $W_{PDS} = 0.679$, $W_{logistic} = 0.618$, $p \leq 0.001$); the differences are significant at a level of $p \leq 0.001$ ($Z = 211$). All pairwise differences between our own approaches and the baseline comparisons are significant at $p \leq 0.001$, as assessed through paired Wilcoxon signed rank tests.

5.1.2. Structural Diversity

The notion of structural diversity has been used to assess connections between users in social recommender systems [22, 23]. Sanz-Cruzado et al. [23] suggest a measure of connective diversity based on the degree distribution of a graph and the Gini index. They argue that a flat or even distribution is indicative of a "distinctive social circle" of a user, whereas a highly

Table 2
Evaluating Degree Gini Complement

algorithm	mean (±SD)
PDS	0.7025 (± 0.0639)
logistic	**0.9010** (± 0.0710)
BaselineOnly	0.6158 (± 0.0050)
NMF	0.5529 (± 0.0728)
NormalPredictor	0.5490 (± 0.0701)

skewed degree distribution indicates that a few individuals form relationships to many others, thus acting as hubs.

We translate this view to our domain of tracks and genres, and argue that a highly skewed degree distribution in a graph containing recommended tracks and genres indicates, in the most extreme case, that all recommended tracks pertain to the same genre, resulting in a star-shaped graph. The other extreme, a completely flat degree distribution, can be viewed as a lattice, where each track is connected to the same number of genres, and vice versa.

The Gini index, originally a gauge of income disparity, has been suggested to evaluate diversity of recommendations [24, 6, 25]; we adopt the *degree Gini complement* (DGC) suggested by Sanz-Cruzado et al. [23] with a slight modification for undirected graphs[3].

Let $G' = (U', V', E')$ denote a graph containing recommended tracks U' and genres V'; $|G'|$ the number of vertices in the graph; $deg(w)$ the degree of a node $w \in G'$; and $|E'|$ the number of edges in the graph. The degree Gini complement is then defined as:

$$DGC(G') = 1 - \frac{1}{|G'| - 1} \sum_{i=1}^{|G'|} (2i - |G'| - 1) \frac{deg(w_i)}{\frac{|E'|}{2}}$$

By using the complement, we ensure that higher values indicate a more even or flat distribution.

For all recommendation lists returned by the employed algorithms, we construct an induced subgraph G' of the bipartite graph G, containing only tracks and vertices appearing as recommendations, and determine the mean DGC across all users (see Table 2).

Logistic scoring appears to produce the most evenly distributed recommendation graphs in terms of degree. Except *NormalPredictor* ($p = 0.547$) all other data is non-normally distributed, according to a Shapiro-Wilk test. Therefore, we again determine statistical significance of the results of logistic scoring against all other algorithms using a Wilcoxon signed rank test. With the exception of PDS ($p \leq 0.001$), none of the observed differences are significant at $p \leq 0.05$.

We thus assess that, while not significantly outperforming the baseline algorithms, logistic scoring tends to produce more diverse recommendations in terms of structural diversity than the simpler discounting approach.

[3]We use $\frac{1}{2}|E'|$ in the denominator to avoid counting edges twice

Table 3
Evaluating Unexpectedness

algorithm	mean (±SD)
PDS	0.1154 (± 0.0427)
logistic	**0.7033** (± 0.0784)
BaselineOnly	0.2172 (± 0.0321)
NMF	0.2225 (± 0.0465)
NormalPredictor	0.2270 (± 0.0453)

5.2. Novelty

Different notions of novelty exist for recommender systems, again on a local, per-user or per-item level, as well as a global level [23, 26]. It has also been argued that a system produces novel recommendations if its output differs from that of a primitive model generating more "expected" recommendations [3].

In this work, we examine novelty both from a user-perspective, as well as in comparison to the baseline algorithms.

5.2.1. Unexpectedness

The notion of unexpectedness lacks a clear definition. Whereas Ge et al. [3] view it as the deviation from a primitive reference system, Castells et al. [6] approach it via user-specific unexpectedness based on item attributes. Thus, a recommended item appears unexpected to the user if it exhibits high distance (low similarity) to their previous experience.

We adopt this measurement of user-specific unexpectedness and determine, for each item in a user's recommendation list R, the mean cosine-distance $cosdist$ to all items they have previously experienced in our dataset, H (their listening history):

$$Unexp(R) = \frac{1}{|R||H|} \sum_{r \in R} \sum_{h \in H} cosdist(r, h)$$

Table 3 lists mean $Unexp$ for all recommendation lists. Logistic scoring seems to return the most unexpected tracks to users. A Shapiro-Wilk test for normality indicated non-normal distributions for all data on the significance level $p \leq 0.001$, hence we test for significance of differences between logistic scoring and the other algorithms using Wilcoxon signed-rank tests, as before. All differences are significant at $p \leq 0.001$.

We attribute the high unexpectedness of results from logistic scoring to the fact that items with low similarity to the user-profile are ranked more highly, providing they also exhibit low popularity.

5.2.2. Jaccard Index

We further assess novelty using the Jaccard index on items produced by the PDS and logistic scoring methods, versus those of the baseline algorithms, which we thus treat as reference

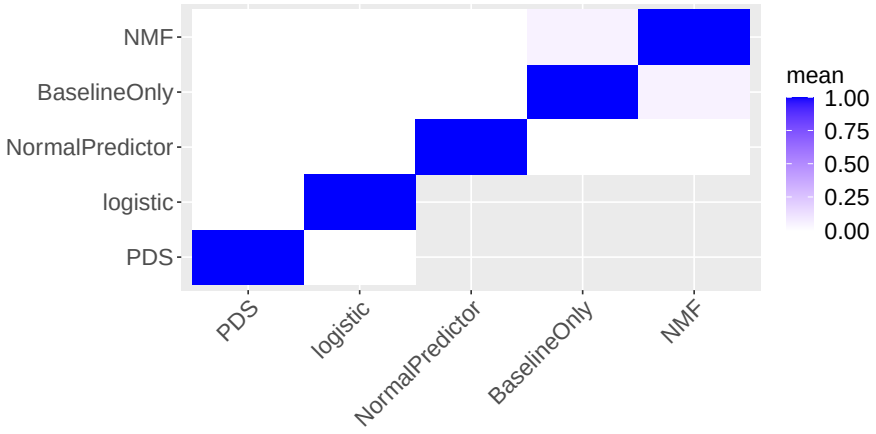

Figure 4: Heatmap of mean Jaccard index for pairs of algorithms.

systems producing "expected" recommendations.

The Jaccard index of two sets A and B is defined as the size of their intersection, normalized by the size of their union:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

To this end, we take the recommendation lists for a user and determine the Jaccard index pairwise on all algorithms, to ensure that we compare the same sets of recommended tracks for a user, and report the mean index value per algorithm. As the heatmap in Figure 4 shows, mean Jaccard indexes tend to be close to 0.00, with the exception of the pair (NMF, BaselineOnly), for which we observe a value of 0.055.

From these results, we conclude neither popularity-discounted similarity nor logistic scoring produces recommendations which would be expected by a baseline algorithm.

5.3. Summary of results

In summary, we observe that logistic scoring, which assigns higher ranks to less popular and more different items, results in a more diverse list of recommendations in terms of Intra-list Diversity than the simpler popularity-discounted similarity. In terms of structural degree-diversity, logistic scoring outperforms all other models, although significance at $p \leq 0.001$ was only observed in comparison to PDS. Logistic scoring also produces the most unexpected recommendations when compared to the other models, with statistical significance. Using the Jaccard index, we determined that none of our scoring functions result in "predictable" recommendations that could be expected from the baseline algorithms.

6. Conclusion

In this work, we constructed a knowledge graph of the musical domain consisting of tracks and their genre-annotations. Using this knowledge graph and contextual information on tracks in the form of acoustic features, we generate a set of recommendation tracks for a sample of 1 000 users drawn from the LFM-1b dataset. We aim to suggest items that tie in with users' previous musical interest, but are tailored towards novelty and diversity, as opposed to reproducing previous experience, thus aiming to foster serendipitous discovery of tracks of music. We compare our results to those of several baseline algorithms and find that while our recommendations exhibit higher novelty in terms of user-specific unexpectedness, the baselines tend to perform better in terms of Intra-list Diversity.

We attribute these results to two reasons: 1), our approach assigns high scores to tracks that are at the same time far from the user's listening profile and unpopular on a global level, thus favouring more novel music with relation to a user's previous encounters; and 2), as we determine candidate tracks for recommendations from pairs of genres the user has already experienced, they are likely to exhibit similar acoustic features, thus exhibiting lower diversity on a per-item level. As an example, consider two tracks being recommended that are both part of the "jazz" and "blues" genres. These tracks are likely more similar in a musical sense than one track from the "blues" and one from the "blackmetal" genre. This is a constraint inherent to our approach, as we aim to suggest novel, but relevant tracks, and ensure relevance by grounding recommendations on genres the user has previously been exposed to.

It must also be noted that the datasets we base our study on may be missing tracks the user has indeed consumed, but that either did not have any (meaningful) genres assigned in the form of *last.fm* tags, or whose acoustic features were not retrievable from the *Spotify*-API. In addition, as applies to many other recommendation settings, the system only has a partial view on users' previous exposure to items. As such, recommendations may well contain content the user is familiar with, but that they have consumed via other on- or offline media.

As we validated our results in an offline setting, it is not possible to reliably state the utility of our recommendations to users, i.e., if they would indeed consider listening to these tracks. To gain deeper insight, an online evaluation in the form of a user-study would be required.

Lastly, although we conducted our work in the domain of music, we believe the ideas extend to other settings, e.g., digital archives. Given an ontology of the domain and a notion of similarity between items, a set of candidates can be ranked to favour those more different to a reference collection. Similarity may be determined in terms of documents' metadata, or latent features, e.g., topic- or sentiment-models.

Acknowledgments

This article is the outcome of research conducted within the Africa Multiple Cluster of Excellence at the University of Bayreuth, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2052/1 – 390713894.

References

[1] C.-N. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proceedings of the 14th international conference on World Wide Web, WWW '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 22–32. doi:10.1145/1060745.1060754.

[2] G. Adomavicius, Y. Kwon, Maximizing aggregate recommendation diversity: A graph-theoretic approach, in: Proc. of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), Citeseer, 2011, pp. 3–10.

[3] M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond accuracy: evaluating recommender systems by coverage and serendipity, in: Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 257–260. doi:10.1145/1864708.1864761.

[4] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino, P. Molino, Introducing serendipity in a content-based recommender system, in: 2008 Eighth International Conference on Hybrid Intelligent Systems, IEEE, 2008. doi:10.1109/his.2008.25.

[5] Y. C. Zhang, D. Ó Séaghdha, D. Quercia, T. Jambor, Auralist: introducing serendipity into music recommendation, in: Proceedings of the fifth ACM international conference on Web search and data mining, WSDM '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 13–22. doi:10.1145/2124295.2124300.

[6] P. Castells, N. J. Hurley, S. Vargas, Novelty and Diversity in Recommender Systems, Springer US, Boston, MA, 2015, pp. 881–918. doi:10.1007/978-1-4899-7637-6_26.

[7] Z. Huang, W. Chung, T.-H. Ong, H. Chen, A graph-based recommender system for digital library, in: Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries - JCDL '02, ACM Press, 2002. doi:10.1145/544220.544231.

[8] A. Tommasel, J. M. Rodriguez, D. Godoy, I Want to Break Free! Recommending Friends from Outside the Echo Chamber, in: Fifteenth ACM Conference on Recommender Systems, Association for Computing Machinery, New York, NY, USA, 2021, pp. 23–33. doi:10.1145/3460231.3474270.

[9] K. Lee, K. Lee, Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items, Expert Systems with Applications 42 (2015) 4851–4858. doi:10.1016/j.eswa.2014.07.024.

[10] S. Chaudhari, A. Azaria, T. Mitchell, An entity graph based recommender system, AI Communications 30 (2017) 141–149. doi:10.3233/AIC-170728.

[11] S. Oramas, V. C. Ostuni, T. D. Noia, X. Serra, E. D. Sciascio, Sound and Music Recommendation with Knowledge Graphs, ACM Transactions on Intelligent Systems and Technology 8 (2016) 21:1–21:21. doi:10.1145/2926718.

[12] M. Schoenfeld, J. Pfeffer, Shortest path-based centrality metrics in attributed graphs with node-individual context constraints, Social Networks (2021). doi:10.1016/j.socnet.2021.10.004.

[13] D. Kowald, P. Muellner, E. Zangerle, C. Bauer, M. Schedl, E. Lex, Support the underground: characteristics of beyond-mainstream music listeners, EPJ Data Science 10 (2021). doi:10.1140/epjds/s13688-021-00268-9.

[14] E. Zangerle, M. Pichl, M. Schedl, User models for culture-aware music recommendation:

Fusing acoustic and cultural cues, Transactions of the International Society for Music Information Retrieval 3 (2020) 1–16. doi:10.5334/tismir.37.

[15] M. Schedl, The LFM-1b dataset for music retrieval and recommendation, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ACM, 2016. doi:http://dx.doi.org/10.1145/2911996.2912004.

[16] M. Schedl, M. Mayr, P. Knees, Music Tower Blocks: Multi-Faceted Exploration Interface for Web-Scale Music Access, in: Proceedings of the 2020 International Conference on Multimedia Retrieval, Association for Computing Machinery, New York, NY, USA, 2020, pp. 388–392. doi:10.1145/3372278.3391928.

[17] C. Leys, C. Ley, O. Klein, P. Bernard, L. Licata, Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median, Journal of Experimental Social Psychology 49 (2013) 764–766. doi:10.1016/j.jesp.2013.03.013.

[18] S. Wasserman, K. Faust, S. U. o. I. W. Urbana-Champaign), Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[19] N. Hug, NicolasHug/Surprise, 2022. URL: https://github.com/NicolasHug/Surprise.

[20] Y. Koren, Factor in the neighbors, ACM Transactions on Knowledge Discovery from Data 4 (2010) 1–24. doi:10.1145/1644873.1644874.

[21] X. Luo, M. Zhou, Y. Xia, Q. Zhu, An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems, IEEE Transactions on Industrial Informatics 10 (2014) 1273–1284. doi:10.1109/TII.2014.2308433.

[22] X. L. Huang, M. Tiwari, S. Shah, Structural diversity in social recommender systems, in: Proceedings of the 5th ACM RecSys Workshop on Recommender Systems and the Social Web, Citeseer, 2013.

[23] J. Sanz-Cruzado, S. M. Pepa, P. Castells, Structural Novelty and Diversity in Link Prediction, in: Companion Proceedings of the The Web Conference 2018, WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 1347–1351. doi:10.1145/3184558.3191576.

[24] D. M. Fleder, K. Hosanagar, Recommender systems and their impact on sales diversity, in: Proceedings of the 8th ACM conference on Electronic commerce, EC '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 192–199. doi:10.1145/1250910.1250939.

[25] A. Gunawardana, G. Shani, Evaluating Recommender Systems, Springer US, Boston, MA, 2015, pp. 265–308. doi:10.1007/978-1-4899-7637-6_8.

[26] S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 109–116. doi:10.1145/2043932.2043955.