

AN INTERPRETABLE MUSIC SIMILARITY MEASURE BASED ON PATH INTERESTINGNESS

Giovanni Gabbolini

Insight Centre for Data Analytics
School of Computer Science & IT
University College Cork, Ireland

giovanni.gabbolini@insight-centre.org

Derek Bridge

Insight Centre for Data Analytics
School of Computer Science & IT
University College Cork, Ireland

d.bridge@cs.ucc.ie

ABSTRACT

We introduce a novel and interpretable path-based music similarity measure. Our similarity measure assumes that items, such as songs and artists, and information about those items are represented in a knowledge graph. We find paths in the graph between a seed and a target item; we score those paths based on their interestingness; and we aggregate those scores to determine the similarity between the seed and the target. A distinguishing feature of our similarity measure is its interpretability. In particular, we can translate the most interesting paths into natural language, so that the causes of the similarity judgements can be readily understood by humans. We compare the accuracy of our similarity measure with other competitive path-based similarity baselines in two experimental settings and with four datasets. The results highlight the validity of our approach to music similarity, and demonstrate that path interestingness scores can be the basis of an accurate and interpretable similarity measure.

1. INTRODUCTION

The concept of similarity is central to music information retrieval, as it underpins important applications like recommender systems and visualisation in user interfaces [1]. In recent years, there has been a surge in the use of knowledge graphs to solve various information retrieval tasks. For example, knowledge graphs have been applied to recommender systems [2] and question answering [3]. In this work, we use knowledge graphs to gauge music similarity. We introduce a novel path-based similarity measure. We represent items, such as songs and their artists, and information about those items in a knowledge graph; we find paths in the graph between a seed and a target item; we score those paths; and we aggregate the scores to determine the similarity between the seed and the target.

We propose to score paths based on interestingness. Interestingness is introduced in [4] as a way of distinguishing

Path (in natural language)	Int.
George Jones was married to Tammy Wynette.	0.60
Tammy Wynette was the parent of the artist Georgette Jones and George Jones was also the parent of the same artist.	0.56
Tammy Wynette wrote “Our Private Life” and George Jones also wrote the same song.	0.45
Tammy Wynette has made country music, and so did George Jones.	0.34
Tammy Wynette was based in United States and George Jones was based in the same country.	0.31
Tammy Wynette was a solo artist, and George Jones was a solo artist.	0.29

Table 1. An example of the working of our proposed similarity measure. In this example, it is given the artists Tammy Wynette and George Jones as seed and target items.

more interesting from less interesting paths in a knowledge graph. In [4], the authors use what they call *segues* to connect a song to the next song, e.g. in a playlist. Segues are translations into natural language of interesting paths in a knowledge graph. We also employ the path-to-text module from [4], useful to translate paths to natural language. In this paper, we are extending [4] by showing that these paths and their interestingness scores can be the basis of an accurate and interpretable similarity measure.

For example, given the artists Tammy Wynette and George Jones as seed and target, our algorithm finds six paths between the two items, then it assigns an interestingness score to the paths, and it aggregates the scores to determine the similarity. It can convert the paths to natural language for display to a user. We report in Table 1 its natural language translations of the paths (Path column) and the interestingness scores (Int. column).

A distinguishing feature of our similarity measure is its interpretability. Miller defines interpretability as the degree to which a human can understand the cause of a decision [5]. According to this definition, our similarity measure is interpretable. Users can understand the causes of a similarity score by looking at natural language translations of the paths used to compute it, ranked by their interestingness. For example, a user can understand why our al-



gorithm thinks that Tammy Wynette and George Jones are similar, when presented with the ranked list of explanations in Table 1. Notice how the interestingness score intuitively reflects the interestingness of the paths. For example, two artists being married has higher score than two artists both being solo singers.

Music similarity measures are usually not interpretable, e.g. see [6]. This may be related to a system-centric research protocol, which is common in the field, and consists of laboratory experiments conducted without end-user involvement, e.g. the evaluation of algorithms on digital databases [7]. This protocol is pragmatic but it may have contributed to widening the “semantic gap” in music similarity measures, which sets a bound on user satisfaction [1]. The term “semantic gap” refers to the discrepancy between high-level perception of similarity by humans and the low-level numerical data used by algorithms [8]. For example, a user might perceive two songs as similar because of their lyrics, while an algorithm might be limited to using abstract statistical descriptors operating on the audio signals. One way of alleviating the semantic gap is the introduction of mid-level features. Mid-level features either combine low-level features or extend them to ones that incorporate additional, higher-level knowledge [1]. Another way of reducing the gap is the adoption of interpretable similarity measures. An interpretable similarity measure can narrow the gap between the low perceptual level of an algorithm, which makes a decision about similarity, and the high perceptual level of a human, by providing an explanation for the human to understand the decision.

The remainder of the paper is organised as follows: in Section 2, we frame our similarity measure in the literature of the subject; in Section 3, we formalise our similarity measure; and, in Section 4, we validate our similarity measure in four experimental conditions. The source code supporting this study is freely available.¹

2. RELATED WORK

The concept of similarity is central to Music Information Retrieval. Over the years, researchers have proposed many ways of measuring music similarity, which can be categorised based on the data they use. We might, for example, categorise them into two: content-based approaches that use the audio signal; and context-based approaches that use information about artists and the pieces of music themselves, e.g. their lyrics, listening logs, and so on [9]. We refer the reader to the book by Knees & Schedl [6] for an extensive review of content-based and context-based approaches. The similarity measure that we propose in this paper is context-based.

Some of the context-based approaches make use of knowledge graphs. These approaches represent entities from the music domain and information about those entities as nodes in a graph, and they use graph structure to gauge similarity. (We define knowledge graphs more formally in Section 3.1.)

Many knowledge graph-based approaches to the measurement of similarity are path-based. They use the existence of paths between the seed and the target entities to determine similarity. Path-based algorithms usually score paths based on some criteria, and then aggregate path scores to determine similarity. For example, in [10] Leal et al. describe Shakti. Shakti scores each path by summing hand-crafted node weights and edge weights, and then dividing by the cube of the path length. Shakti only considers paths whose length falls below a fixed threshold. In [11], Strobin & Niewiadomski extend Shakti by setting node and edge weights using a genetic algorithm. This extension evolves different sets of weights indexed by path length, and considers paths of any length. They adjust node weights by considering centrality, i.e. the number of in-going and out-going edges of a node.

Passant proposes another path-based algorithm, called LDSO [12]. LDSO considers only paths of length up to two. It assigns low scores to paths whose type is common in the context of the seed and target entities. Piao & Breslin extend LDSO in [13]. They propose four variants of LDSO, which correct the path scoring mechanism with global information, such as the frequency of path types in the whole knowledge graph. In a comparative study, Piao et al. show that LDSO largely outperforms Shakti in accuracy [14].

The similarity measure that we propose in this paper is also path-based. It is different from the works in the literature as we employ a novel scoring mechanism for paths, based on their interestingness.

As well as path-based approaches to the measurement of similarity, there are also embedding-based approaches. These work by representing the knowledge graph in a dense, low-dimensional feature space, which tries to preserve as much of the graph’s structural information as possible. There are several ways to learn knowledge-graph embeddings. We refer the reader to [15] for a survey. Once the embeddings are learned, the similarity between entities can be computed on their vector representations using, for example, cosine similarity.

Path-based and embedding-based approaches to the measurement of similarity differ in their interpretability. In [16], Du et al. provide a characterization of approaches based on their potential for interpretability. They introduce the concept of *intrinsic interpretability*, which is achieved by constructing self-explanatory models. Path-based approaches typically exhibit intrinsic interpretability, since the paths used to gauge similarity can be considered to be self-explanatory. Embedding-based approaches do not exhibit intrinsic interpretability, since the embeddings used to gauge similarity cannot be considered to be self-explanatory. As a consequence, path-based approaches are interpretable after the translation of the paths to natural language, while embedding-based approaches can become interpretable only after the introduction of a post-hoc model, built to generate explanations [16].

Path-based approaches are not the only category of similarity measures to feature intrinsic interpretability. For ex-

¹ <https://github.com/GiovanniGabbolini/ipsim>

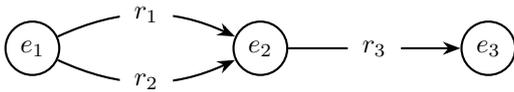


Figure 1. A simple knowledge graph.

ample, if items are described by sets of tags, then Jaccard similarity over tags also exhibits intrinsic interpretability. The similarity can be explained by showing the tags shared between the seed and the target. Even so, we find a scarcity of interpretable similarity measures in the literature; the book [6] contains very few, for example.

We mention that Passant’s LDS is also interpretable, since Passant implements a mechanism to translate paths to natural language [12].

3. METHOD

As we have already indicated, our approach to the measurement of similarity of entities in knowledge graphs is path-based. It builds on work described in [4], where the authors introduce interestingness as a way of distinguishing more interesting from less interesting paths in a knowledge graph.

In the following, we first introduce some preliminary concepts, then we review [4]’s definition of interestingness, and finally we introduce our similarity measure.

3.1 Preliminaries

Both our work and [4] make use of a knowledge graph G as an abstract representation for items and information about those items. In the experiments that we report later in this paper, items are artists; in [4], items are songs. But the approach is domain-independent: items can be any entities of interest.

A **knowledge graph** is a set of triples $G = \{(e, r, e') \mid e, e' \in E, r \in R\}$, where E and R denote, respectively, the sets of entities (nodes) and relationships (edges). For example, the knowledge graph of Figure 1 contains three triples, $\{(e_1, r_1, e_2), (e_1, r_2, e_2), (e_2, r_3, e_3)\}$. A special subset of entities $I \subseteq E$, are the items. Every entity has a *type* and a *value*. For example, if an entity represents an artist, then its type is “artist” and its value is the URI of that artist. Every edge (relationship) has a *type* also. For example, an edge that connects two artists who have collaborated will have “collaborated with” as its type.

A **path** p in G is an ordered list of entities and edges in G , $p = [e_1, r_1, \dots, r_{n-1}, e_n]$ where each triple in p must be in G . For example, with reference to the knowledge graph of Figure 1: $[e_1, r_1, e_2, r_3, e_3]$ is a path; $[e_1, r_1, e_2, r_1, e_3]$ is not a path, as the triple (e_2, r_1, e_3) is not in the knowledge graph. The *type* of p is the ordered concatenation of the entity and edge types in p .

The function $paths(i_1, i_2)$ finds the paths in G from the item $i_1 \in I$ to the item $i_2 \in I$, without visiting any other item in I and without cycles. In other words, if items are songs, for example, then the function finds paths between pairs of songs, where the paths are not allowed to contain any intermediate songs.

3.2 Interestingness

In [4], the interestingness of a path in a knowledge graph is defined as a weighted combination of three heuristics:

Rarity Let T be the set of all path types in G ; and let $f(t)$ be the number of paths in G that are of type t . The *rarity* of a path p is defined as:

$$rarity(p) = 1 - \frac{f(type(p))}{\max_{t \in T} f(t)}$$

Unpopularity Let $edgeset(e)$ be the set of in-going and out-going edges to and from an entity $e \in E$ in G . The *centrality* of an entity e is:

$$centrality(e) = \min \left(1, \frac{|edgeset(e)|}{\text{median}_{e' \in E} |edgeset(e')|} \right),$$

given $type(e') = type(e)$

The *unpopularity* of a path p is defined as:

$$unpopularity(p) = 1 - \min_{e \in p \cap E} (centrality(e))$$

Shortness If we define $length(p)$ as the number of edges in path p , then the *shortness* of a path is given by:

$$shortness(p) = \frac{1}{length(p)}$$

The *interestingness* of a path p in G is given by:

$$interestingness(p) = w_1 rarity(p) + w_2 unpopularity(p) + w_3 shortness(p)$$

Its values range from zero to one. w_1, w_2, w_3 are parameters to be tuned, subject to $w_1 + w_2 + w_3 = 1$.

In [4], the authors run a user trial where participants are asked to evaluate segues. They find that *interestingness* positively correlates with human perceptions of quality: *interestingness* is high for segues perceived as high-quality, and vice versa. This lends credence to the idea of using their definition of interestingness in the similarity measure that we are proposing in this paper.

3.3 Similarity

We define the similarity between two items $i_1 \in I$ and $i_2 \in I$ as an aggregate of the interestingness of the paths between them:

$$sim(i_1, i_2) = \sum_{p \in paths(i_1, i_2)} interestingness(p) \mathcal{I}(p, n)$$

where \mathcal{I} is an indicator function: $\mathcal{I}(p, n) = 1$ if $\text{length}(p) \leq n$, and 0 otherwise. n is a parameter whose value determines whether the aggregation restricts attention to shorter paths or whether it considers all paths ($n = \infty$). It is not clear *a priori* whether it is beneficial to put a limit on path length and, if so, what limit to use. For example, [12] use only paths of length up to three, while [11] uses all paths. In our case, we leave the choice open by introducing the parameter n to be tuned. The interestingness weights w_1, w_2, w_3 are also parameters to be tuned.

In the following, we refer to our similarity measure as **IPSIM (Interesting Path Similarity)**. IPSIM is domain-independent. In the rest of this paper, we focus on the music domain, and in particular the case where items are music artists. The reason we focus on artist similarity is that it is a common choice in the literature and there is an amount of data to use in offline experiments.

4. EXPERIMENTS

We provide an evaluation for IPSIM in the case of music artists. In this evaluation, we are particularly interested in evaluating the accuracy that IPSIM is able to achieve.

4.1 Knowledge graph

We represent an artist with their MusicBrainz URI.² This representation can be easily changed with only minor modifications to the rest of our implementation, e.g. if artists were instead represented by their Spotify URIs.

Our implementation uses a knowledge graph with 30 distinct node types and 205 distinct edge types. It is built with data that we harvest from two resources:

MusicBrainz We use MusicBrainz as the main source of factual data. We exploit the MusicBrainz APIs.³ They allow us to navigate the MusicBrainz database, and offer entity-linking functionalities. We mine different sorts of factual data, ranging from the birth places of the artists to the genres that they play.

Wikidata We use Wikidata as an additional source of factual data.⁴ There exists a mapping from MusicBrainz URIs to Wikidata URIs, making it easy to use both resources. From Wikidata, we mine biographical data about artists that is not available in MusicBrainz, e.g. the awards that an artist has won.

We provide a complete description of entities and relationships that build up the knowledge graph that we use in this paper in the additional materials.⁵

4.2 Experiment design

As highlighted by Knees & Schedl [1], there is no agreement upon the best method to evaluate the quality of mea-

asures of music similarity. Every method has its own advantages and disadvantages. In this paper, we assess music similarity measures using two common experimental settings. The first uses a similar-artists ground-truth, and the second uses user-artist interaction histories.

4.2.1 Using a ground-truth

This evaluation procedure replicates the one in [17]. The procedure uses datasets that are composed of tuples of the form (*seed artist, similar-artists ground-truth*). In other words, for each seed artist, it gives a list of artists that have been independently judged to be similar to the seed, these artists being regarded as the ground-truth. We employ two datasets:

MIREX: This dataset comprises 188 seed artists. The ground truth is based on similarity judgements expressed by experts during the MIREX Audio Music Similarity and Retrieval Task.

LastFM-g: This dataset comprises 2336 seed artists. The ground truth is based on similarity judgements gathered from the Last.fm APIs.⁶

See [17] for additional details on these datasets.

To evaluate a similarity measure on a given dataset, we do the following. For every seed artist, we use the similarity measure to score the other artists and then to rank them by decreasing similarity with the seed. Then, we compare this ranking @ N with the ground-truth from the dataset, according to the standard accuracy metrics *nDCG* and *Precision*, or *Pr* for short.

4.2.2 Using interaction histories

This protocol uses a dataset that records how users have interacted with artists. We denote with U, I the sets of users and artists resp. and with $|U|, |I|$ their cardinalities. Lower case letters u, i refer to $u \in U, i \in I$. In this case, the dataset is organised as a matrix $R \in \mathbb{R}^{|U| \times |I|}$. Each cell $r_{ui} > 0$ of R accounts for a user-artist interaction. If a user u is not known to have interacted with an artist i , then r_{ui} is zero.

We employ two datasets of this kind:

LastFM-h: In this Last.fm dataset (which is different from the one described earlier), r_{ui} is the listening count by user u of artist i [18].⁷ We filter out users who have listened to fewer than five artists, and artists listened to by fewer than five users. We end up with 1877 users and 2828 artists. We convert listening counts to ratings on a 1–5 scale following the procedure given in [19]. For every user, we hold-out at random 40% of her interactions, and we keep the other interactions as training data. We further split the held-out interactions into two equal-sized parts, to form validation and test data. Finally, we remove the interactions where $r_{ui} < 4$ from the validation and test data so that they only contain artists that the users like.

² <https://musicbrainz.org/>

³ <https://python-musicbrainzngs.readthedocs.io/en/v0.7.1/>

⁴ <https://wikidata.org/>

⁵ <https://doi.org/10.5281/zenodo.5121460>

⁶ last.fm

⁷ <https://grouplens.org/datasets/hetrec-2011/>

Facebook: This dataset, from the Second Linked Open Data-enabled Recommender Systems Challenge, contains items liked by users in three domains: movies, books and music.⁸ We focus on music artists, by keeping only items of types: *music_artist* and *music_band*. We filter out artists liked by fewer than five users, and those whose MusicBrainz URI cannot be resolved. We end up with 52069 users and 4435 artists. In this case, $r_{ui} = 1$ if the user u likes the artist i . For every user, we hold-out 40% of her interactions at random, and we keep the others as training data. We further split the held-out interactions into two equal-sized parts, to form validation and test data.

For these two datasets, the evaluation procedure uses the similarity measure as a core component within a recommender system. The design of the recommender system is inspired by [13], chosen because of its heavy reliance on similarity, which is what we are trying to evaluate. We define the user profile of a user u as the artists she has rated (in the training set):

$$P(u) = \{j | r_{uj} \neq 0\}.$$

Let $NN(k, i, P(u))$ be the set of k most similar items in $P(u)$ to an artist i . The predicted relevance score for a user u and an artist i , indicated as $\overline{r_{ui}}$, is computed as:

$$\overline{r_{ui}} = \begin{cases} 0 & \text{if } i \in P(u) \\ \sum_{j \in NN(k, i, P(u))} r_{uj} \text{ sim}(i, j) & \text{if } i \notin P(u) \end{cases}$$

For every user, we compute the relevance score $\overline{r_{ui}}$ for every artist $i \in I$, and rank the artists by decreasing score. Then, we compare this ranking @ N and the held-out test data (or validation data, as appropriate), according to the standard accuracy metrics $nDCG$ and *Precision*, or *Pr* for short. We take higher recommendation accuracy to be an indication of a higher quality similarity measure.

4.3 Baseline similarity measures

We include in our experiments three baselines. The first is a random algorithm (RND), useful to set a lower bound on the performance. It rates the similarity between two items to be a random number between zero and one. The other two (COUNT and LDSD) are path-based algorithms. By comparing our path-based approach to other path-based similarities, we reduce the number of confounders, so we can investigate whether our idea of using interestingness as a scoring mechanism for paths is a good one. In fact, the path-based similarity measures that we use in this experiment all use the same knowledge graph and the same paths, and differ only in the way the paths are scored and how the scores are aggregated. COUNT is a simple path-based algorithm that works by counting the number of paths between items. It is equivalent to substituting 1 for the argument to the sum in the definition of *sim* given earlier. LDSD is an accurate path-based algorithm that proposes a carefully-designed weighting heuristic for paths [12].

⁸<https://lists.w3.org/Archives/Public/public-vocabs/2015Feb/0046.html>

	w_1	w_2	w_3	n	k
<i>MIREX</i>	0.3	0.1	0.6	2	NA
<i>LastFM-g</i>	0.0	0.1	0.9	2	NA
<i>LastFM-h</i>	0.0	0.0	1.0	2	40
<i>Facebook</i>	0.9	0.0	0.1	∞	∞

Table 2. Optimal parameters found for IPSIM.

4.4 Parameter tuning

IPSIM has some parameters, as indicated in Section 3.3. Additionally, the experiment using listening histories introduces another parameter, i.e. the number of neighbours k used by the recommender system, as described in Section 4.2.2. We set the parameters with a grid-search, optimizing $nDCG@10$ on the validation data. For the experiment using a ground-truth, we tune the parameters of IPSIM. For the experiment using listening histories, we tune the parameters of IPSIM and we tune k for each of RND, COUNT, LDSD and IPSIM. We report the optimal parameters found for IPSIM in Table 2. The optimal values of the parameter k for RND, COUNT and LDSD are, respectively, 5, 10 & 40 in *LastFM-h* and 1, 1 & 7 in *Facebook*.

The optimal parameter configurations change from dataset to dataset. This might be due to the different nature of the datasets. *MIREX* features human judgements of artist similarity; the algorithmic judgments to be found in *LastFM-g* might not completely agree with these. Schedl et al. [7] suggest that human perception of similarity is influenced by user factors, such as user context (e.g. mood) and user properties (e.g. musical training). User factors might have directly influenced the people who annotated *MIREX*, but not the algorithm used to construct *LastFM-g*. The remaining two datasets, *LastFM-h* and *Facebook*, gather another kind of data, i.e. interaction histories, but they are gathered in different ways. In *LastFM-h*, unless the user intervenes, the next recommended song is played automatically, whereas, in *Facebook*, users are interacting with artist fan pages in what is usually a more deliberate, conscious and considered way.

In three of the four datasets, at least one of the three interestingness heuristics of IPSIM receives a weight of zero. We may find an explanation for this in the median number of paths between artists: 19 in *MIREX*, 13 in *LastFM-g*, and seven in both *LastFM-h* and *Facebook*. *MIREX* is the only dataset where all three weights are non-zero, and is also the dataset with the highest median number of paths between artists. In general, the use of a more complex model (with non-zero weights for each component) might be beneficial only when the dataset refers to artists in the knowledge graphs for whom there exists a rich variety of paths, e.g. when considering popular artists.

4.5 Results

We conduct the experiments described in Section 4.2. We compare IPSIM against the baselines described in Section 4.3 (RND, COUNT, LDSD). We set the parameters of the similarity measures and the recommender as described in

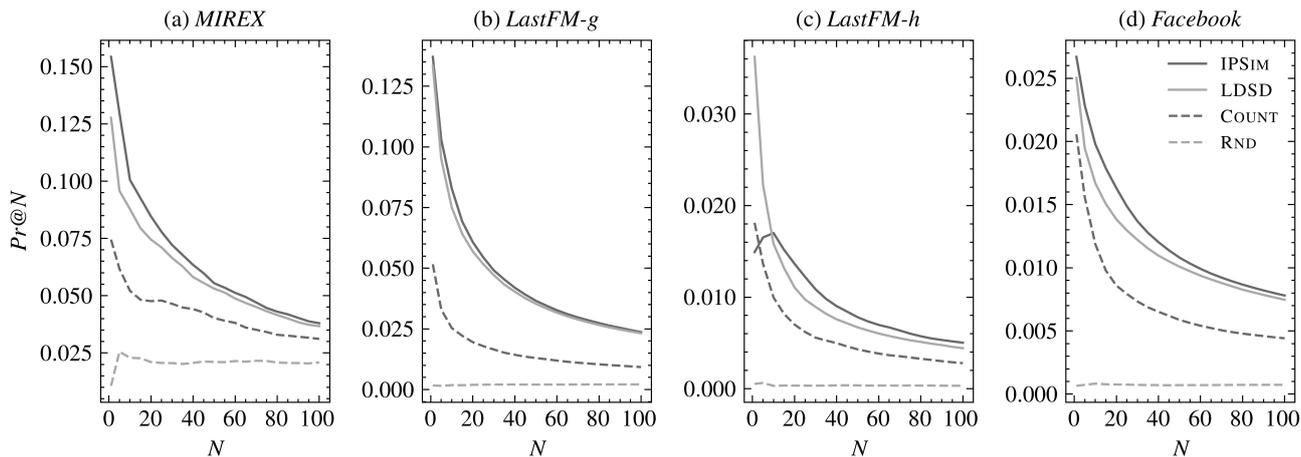


Figure 2. Accuracy of similarity measures on the *MIREX*, *LastFM-g*, *LastFM-h* and *Facebook* datasets, as measured by *Precision* at cutoffs from one to 100.

	<i>nDCG@5</i>	<i>nDCG@10</i>	<i>Pr@5</i>	<i>Pr@10</i>
RND	0.023	0.022	0.026	0.023
COUNT	0.065	0.057	0.062	0.052
LDSD	0.102	0.094	0.096	0.088
IPSIM	0.136***	0.114*	0.130***	0.101*

Table 3. Accuracy of similarity measures on the *MIREX* dataset, as measured by *nDCG* and *Precision* at cutoffs five and ten. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

	<i>nDCG@5</i>	<i>nDCG@10</i>	<i>Pr@5</i>	<i>Pr@10</i>
RND	0.002	0.002	0.002	0.002
COUNT	0.037	0.030	0.033	0.025
LDSD	0.103	0.086	0.095	0.075
IPSIM	0.111**	0.094***	0.103**	0.083***

Table 4. Accuracy of similarity measures on the *LastFM-g* dataset, as measured by *nDCG* and *Precision* at cutoffs five and ten. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

	<i>nDCG@5</i>	<i>nDCG@10</i>	<i>Pr@5</i>	<i>Pr@10</i>
RND	0.001	0.000	0.001	0.000
COUNT	0.015	0.012	0.014	0.010
LDSD	0.025***	0.020**	0.022***	0.016
IPSIM	0.016	0.017	0.017	0.017

Table 5. Accuracy of similarity measures on the *LastFM-h* dataset, as measured by *nDCG* and *Precision* at cutoffs five and ten. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

	<i>nDCG@5</i>	<i>nDCG@10</i>	<i>Pr@5</i>	<i>Pr@10</i>
RND	0.001	0.001	0.001	0.001
COUNT	0.017	0.014	0.016	0.012
LDSD	0.021	0.018	0.019	0.017
IPSIM	0.024***	0.021***	0.023***	0.020***

Table 6. Accuracy of similarity measures on the *Facebook* dataset, as measured by *nDCG* and *Precision* at cutoffs five and ten. *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

Section 4.4 using the validation data. We report the results on the test data. We verify the significance of differences between IPSIM and LDSD with Wilcoxon signed-rank test.

Tables 3 & 4 report on the experiment that uses a ground-truth. IPSIM outperforms all the baselines in both the *MIREX* and *LastFM-g* datasets. We notice that IPSIM scores accuracy at least double that of COUNT in both datasets. The increase in performance with respect to LDSD is statistically significant in both datasets.

Tables 5 & 6 report on the experiment that uses listening histories. IPSIM is always more accurate than COUNT. IPSIM always outperforms LDSD in *Facebook* but outperforms LDSD in only one case in *LastFM-g*, and the difference in this case is not statistically significant.

We investigate more deeply how accuracy varies with the cutoff in Figure 2. For the four datasets, the figure shows the *Precision* of the similarity measures as a function of the cutoff. The figure confirms that IPSIM has higher precision than LDSD on *LastFM-h* for cutoffs greater than ten, but the precision of IPSIM drops for cutoffs < 10 . We also notice that, on *MIREX*, *LastFM-g* and *Facebook*, IPSIM outperforms LDSD for every cutoff ranging from one to 100.

The results provide evidence that interestingness can be the basis of an accurate similarity measure.

5. CONCLUSIONS AND FUTURE WORK

The results of the experiments highlight the validity of our approach to music similarity (at least in the case of music artists), and demonstrate that the interestingness scores of [4] can be the basis of an accurate and interpretable similarity measure. Future work might include a comparison of the performance of our similarity measure against an even wider range of music similarity measures. We are also interested in using the proposed similarity measure in a related-item recommender system, that can meaningfully guide users in their exploration of the items by means of the interpretability of the similarity measure.

6. ACKNOWLEDGMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

7. REFERENCES

- [1] P. Knees and M. Schedl, "Introduction to music similarity and retrieval," in *Music Similarity and Retrieval: An Introduction to Audio- and Web-based Strategies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 1–30. [Online]. Available: https://doi.org/10.1007/978-3-662-49722-7_1
- [2] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.
- [3] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 615–620. [Online]. Available: <https://www.aclweb.org/anthology/D14-1067>
- [4] G. Gabbolini and D. Bridge, "Generating interesting song-to-song segues with dave," in *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 98–107. [Online]. Available: <https://doi.org/10.1145/3450613.3456819>
- [5] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>
- [6] P. Knees and M. Schedl, in *Music Similarity and Retrieval: An Introduction to Audio- and Web-based Strategies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. [Online]. Available: <https://doi.org/10.1007/978-3-662-49722-7>
- [7] M. Schedl, A. Flexer, and J. Urbano, "The neglected user in music information retrieval research," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 523–539, 2013. [Online]. Available: <https://doi.org/10.1007/s10844-013-0247-6>
- [8] D. Ponceleón and M. Slaney, "Multimedia information retrieval," in *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Professional, 2011, pp. 589–590.
- [9] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, Dec. 2013. [Online]. Available: <https://doi.org/10.1145/2542205.2542206>
- [10] J. Leal, V. Rodrigues, and R. Queirós, "Computing semantic relatedness using DBPedia," *Symposium on Languages, Applications and Technologies, 1st*, no. January, pp. 133–147, 2012.
- [11] Ł. Strobin and A. Niewiadomski, "Evaluating semantic similarity with a new method of path analysis in RDF using genetic algorithms," *Journal of Applied Computer Science*, vol. 21, no. 2, pp. 137–152, 2013.
- [12] A. Passant, "Measuring semantic distance on linking data and using it for resources recommendations," *AAAI Spring Symposium - Technical Report*, vol. SS-10-07, pp. 93–98, 2010.
- [13] G. Piao and J. G. Breslin, "Measuring semantic distance for linked open data-enabled recommender systems," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 315–320. [Online]. Available: <https://doi.org/10.1145/2851613.2851839>
- [14] G. Piao, S. S. Ara, and J. G. Breslin, "Computing the semantic similarity of resources in dbpedia for recommendation purposes," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9544, pp. 185–200, 2016.
- [15] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, 2020. [Online]. Available: <https://www.mdpi.com/2079-9292/9/5/750>
- [16] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, p. 68–77, Dec. 2019. [Online]. Available: <https://doi.org/10.1145/3359786>
- [17] S. Oramas, M. Sordo, L. Espinosa-Anke, and X. Serra, "A semantic-based approach for artist similarity," in *16th International Society for Music Information Retrieval Conference*, Málaga, Spain, 26/10/2015 2015, pp. 100–106. [Online]. Available: <http://hdl.handle.net/10230/26278>
- [18] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, ser. RecSys 2011. New York, NY, USA: ACM, 2011.
- [19] Ò. Celma, "Music recommendation and discovery in the long tail," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2008.