SMARTER THAN GENIUS?
HUMAN EVALUATION OF MUSIC RECOMMENDER SYSTEMS.

Luke Barrington **Reid Oda**[*] **Gert Lanckriet**
Electrical & Computer Engineering, [*]Cognitive Science
University of California, San Diego
lukeinusa@gmail.com roda@ucsd.edu gert@ece.ucsd.edu

ABSTRACT

Genius is a popular commercial music recommender system that is based on collaborative filtering of huge amounts of user data. To understand the aspects of music similarity that collaborative filtering can capture, we compare Genius to two canonical music recommender systems: one based purely on artist similarity, the other purely on similarity of acoustic content. We evaluate this comparison with a user study of 185 subjects. Overall, Genius produces the best recommendations. We demonstrate that collaborative filtering can actually capture similarities between the acoustic content of songs. However, when evaluators can see the names of the recommended songs and artists, we find that artist similarity can account for the performance of Genius. A system that combines these musical cues could generate music recommendations that are as good as Genius, even when collaborative filtering data is unavailable.

1. INTRODUCTION

The popularity of the online radio station Pandora.com (20 million users) and Apple iTunes' "Genius" feature (released in September 2008 and available to over 10 million registered iTunes users) has brought the perennial MIR research topic of music similarity and recommender systems into the public spotlight. Apple, the largest music retailer in the world, collects massive amounts of data about music purchase and listening habits of its users. Our experiments demonstrate that collaborative filtering of this data allows Genius to produce better music recommendations than systems based on simple metadata- or content-based analysis. However, Genius fails on music for which collaborative filtering data is unavailable, such as the huge volume of undiscovered content in the "long tail" of the music market.

In this paper, we seek to understand the musical cues that Genius' collaborative filtering identifies to capture music similarity. We can then develop MIR recommender systems that use the same cues, without the need for massive amounts of user data. Since we do not have access to the collaborative filtering input to the Genius algorithm, we compare its output to two canonical recommender systems where we have complete knowledge of their available musical information. We discover that, despite not basing its recommendations directly on the audio content, collaborative filtering can capture information about acoustic similarity, as well as metadata similarity, for playlist *generation*. Using a blind user study, we determine the influence of certain metadata (e.g., familiarity, affinity, visibility) and musical factors (e.g., styles, sounds, artists) on playlist *evaluation* .

2. THE BLACK ART OF PLAYLIST GENERATION

A playlist is a collection of songs grouped together under a particular principle. The principle could be general, such as "rock songs from the 70's" or personal like "songs that remind me of Melanie". Cunningham et al.[1] make the distinction between playlists and "mixes". While a mix can have abstract themes and the sequence of songs is important, a playlist simply embodies a mood or desired emotional state or acts as a background to an activity (work, romance, sports, etc.). The order of songs in a playlist is not important and it is often played on shuffle. Cunningham et al.'s user study reports that 50 percent of requests for help in creating a playlist included a song as an example. Our work focuses on this "query by example" paradigm where the user provides a song as a query or "seed" and the recommender system's task is to generate a playlist of more music that somehow "fits well" with the seed song. The meaning of "fits well" may depend on a variety of the factors below.

2.1 Factors that impact playlist generation

Playlists may be generated (either automatically or by hand) to reflect a *mood*, accompany an *activity* or explore *novel songs* for music discovery. Recommendations can be based on similarity to one or more seed examples or songs may be grouped based on semantic descriptions. The top organization schemes for playlists in [1] were similar *artists*, *genres* and *styles* so we focus on the impact of these factors for automating playlist generation.

2.2 Factors that influence playlist evaluation

It is rare that a playlist is rated explicitly by the conditions used to generate it. The playlist's purpose plays a large role in evaluating it. Since music is often experienced within a social context[2], factors such as song *popularity*, *familiarity* and the perception of the recommender system as an *expert* can play a large role in the perceived quality of the playlist. Even systems that generate novel or serendipitous playlists for song discovery must include some familiar and relevant items to inspire users to trust the recommender system[3]. This may be achieved by offering some *transparency* of the recommendations, e.g., by showing matching artists or using descriptive tags.

3. MUSIC RECOMMENDER SYSTEMS

A variety of approaches to music recommendation and playlist generation have been proposed by the MIR community. Aucouturier and Pachet [4] used acoustic similarity to group songs together. Flexer et al. [5] propose using KL divergence between acoustic song models to make a playlist that transitions coherently from a start to an end song. Xiao et al.[6] describe songs' acoustic content using automatically generated tags drawn from a variety of semantic categories. They derive a music similarity metric by learning the optimum weighting of these categories and find genre similarity to be the most important predictor of subjective evaluations.

Fields et al.[7] extract social-network flow between artists on MySpace and use the resulting artist association metadata to build playlists. Vignoli and Pauws [8] designed a recommender system that allows users to control how acoustic timbre information is combined with genre, mood, year and tempo metadata. The resulting playlists rated higher than less transparent controls in a user evaluation.

3.1 Two Types of Recommender System

Section 2 details a variety of influences that may be used by music recommender systems but they can be broadly categorized into two different approaches:
Content-based systems "listen" to the audio content of the music and build playlists by finding songs that sound similar (e.g., [4, 5]) or that have similar semantic descriptions (e.g., [9, 6]). For example, the popular online radio station Pandora.com [1] employs professional musicologists to listen to each of the 1 million songs in its "music genome" database and objectively characterize their acoustic content using 400 semantic descriptors (e.g., major or minor tonality, the amount of syncopation, the gender of the vocalist, etc.).
Metadata-based systems use information associated with the music that is not directly related to the acoustic content such as artist names (e.g., [7]), genre or other tag information, purchase data, popularity, etc. For example, the Genius playlist algorithm uses collaborative filtering based on the purchase history of millions of iTunes users (i.e., listeners who bought *this* song also bought *that* song).

For this paper, we evaluate the Genius recommender system against one content-based and one metadata-based approach to generating playlists, as well a system that generates playlists randomly. All systems take a seed song and return a playlist of five recommended songs. Each algorithm that we consider is described in detail below.

3.2 Genius

The iTunes Genius recommender system [2] uses the Gracenote MusicID service[10] to fingerprint songs in a user's music library and identify the name of the song, artist, album, etc. This metadata is then used to identify the songs in Genius' database. Although the exact details of the algorithm are a trade-secret of Apple Inc., Genius appears to use collaborative filtering to compare the seed song's metadata to iTunes' massive database of music sales (over 50 million customers who have purchased over 5 billion songs), as well as play history and song rating data collected from iTunes users [3] . When it is first initialized, Genius analyzes a user's music library and compiles all of the collaborative filtering data necessary to build playlists from the library, based on any given seed song. While this fingerprinting and database communication takes some time ($\sim$ 1 hour for our 12,000-song library), the only acoustic analysis involved seems to be fingerprinting for the purpose of metadata information and not content-based recommendation.

Informal experiments with Genius give some clues into its operation and verify that it does not use content analysis directly. For example, if we delete the ID3 metadata information associated with a given MP3 file, or add a song to the library which is unknown to Gracenote (e.g., a new recording by an obscure band), Genius fails to recommend any music. Furthermore, if we choose a seed song that is very atypical of the style of the artist or album that features the song, Genius recommends music that represents the more common aspects of the artist. For example, using the seed song "Beautiful World", a country-folk ballad that is an outlying anomaly on the album "Renegades" by the metal band "Rage Against the Machine", Genius recommends a playlist of aggressive, thrash-metal songs by bands such as "Incubus" and "Nirvana". Although these artists are related to the seed artist, the sound and style of the resulting playlist is very dissimilar to that of the seed song. Based on this analysis, we expect Genius to perform well when recommending playlists based on popular seed songs but to suffer when analyzing less well-known music.

3.3 Artist Similarity

To provide a second, more transparent playlist algorithm that, like Genius, is not based on acoustic analysis, we consider building playlists based on artist similarity. The social music-streaming website last.fm offers lots of user-generated information about songs and artists [4] . In particular, for any given artist, our artist similarity system retrieves a ranked list of the 100 most similar artists to the seed song's artist. We use this last.fm metadata to build a playlist by moving down the ranked list and choosing a random song by each artist that we find in our library. Comparisons between these music recommendations and Genius' playlists will illuminate the degree to which collaborative filtering captures **artist similarity**.

3.4 Semantic Similarity of Automatic Tags

To examine Genius' ability to capture **acoustic similarity** between songs, we compare it to a purely content-based approach. This recommender system is modeled on Pandora.com in that it finds similar songs by matching semantic

[1] `www.Pandora.com`

[2] Our experiments use Genius incorporated in iTunes version 8.0.

[3] Based on `http://www.apple.com/pr/` as well as a meeting between the authors and iTunes in January 2009.

[4] `www.last.fm/api`

descriptions of the audio content. Pandora's semantic data and its music library are proprietary, so we recreate a similar system using computer audition.

We use an automatic tagging algorithm, described in detail in [11], to describe any song using 149 different semantic tags. These tags include descriptors of the genre, emotion, instruments, vocals and usages of the song. For a given song, the output of this "auto-tagger" is a set of probabilities that indicate the relevance of each tag to the song. These probabilities may be interpreted as the parameters of a "semantic" multinomial distribution that characterizes the song, just as a human listener might use words to describe a song's acoustic content (e.g., "very jazzy, features a lot of saxophone and piano, and good to listen to on a date").

The auto-tag system computes similarity between two songs by comparing the Kullback-Liebler (KL) divergence between their semantic multinomial distributions. To build a playlist, we return songs with minimum KL-divergence from the seed song. Abstracting multimedia representations using semantics has shown improvements over direct feature-based similarity for retrieval of images[12], video[**?**] and sound effects[9] and this system was among the top four performing algorithms in the 2007 MIREX audio similarity challenge [13].

4. PLAYLIST EVALUATION EXPERIMENT

One of the biggest challenges when designing music recommendation systems lies in evaluating any proposed method. There is no standard "ground truth" data set on which to test, let alone train, music similarity algorithms. Widely available surrogates for similarity exist, such as deciding that songs should be deemed "similar" if they come from similar genres [6, 14], artists [15] or albums [9]. Playlists can be evaluated by examining their intersection with existing, human-generated playlists [16, 6] but this requires that the same music libraries be used to generate both the new and the reference playlists.

A more accurate, but less scalable or flexible approach uses humans to evaluate music recommendations. This was the approach taken in the 2007 MIREX contest[13] and, though great effort was required to collect this information, the resulting evaluation was very rich. This data has not been released to the MIR community. Human computation games such as Tag-A-Tune[17] may provide another source of human-derived music similarity data.

Since the goals of this paper are both to evaluate the performance of different music recommender systems in various simulated scenarios *and* determine the factors that influence these evaluations, we built a new platform for humans to evaluate playlists as well as collect information about the strengths and weaknesses of each system.

4.1 The Interface

A new subject arriving at the experiment website sees brief instructions explaining the task and the playlist evaluation procedure. The subject then logs in, so that they can return to the experiment at a later date and not repeat trials.

A single evaluation or "trial" consists of three stages: 1) Listen to and evaluate a seed song. 2) Listen to and evaluate 2 playlists. 3) Indicate factors that influenced the playlist evaluation. 50 seed songs were chosen in advance and, on each trial, one seed song is randomly assigned to a subject (without repetition). In the first stage, the subject listens to the seed song and rates how *familiar* they are with the song and how much they *like* the song, both on a 5-point scale.

Once the subject rates the seed song, stage 2 displays two playlists, each containing 5 songs, generated by one of the 4 possible recommender systems (Genius, Artist Similarity, Similar Tags or a random playlist). The two systems in a given trial are randomly chosen but not the same. The subject can listen to the songs from each playlist in any order or re-listen to the seed song by pressing corresponding play buttons. Beside each song is a button to indicate any bad song that "doesn't fit" in the playlist. After listening to the playlists, the subject evaluates which playlist is better, on a 5-point scale: "Playlist 1 much better", "Playlist 1 somewhat better", "Equal", "Playlist 2 somewhat better", "Playlist 2 much better".

After choosing the winning playlist, stage 3 asks the subject to indicate factors that influenced their evaluation. Six factors are presented that may have affected the subject's evaluation of why *either* playlist was a good match with the seed song: similar **sounds, genres, artists, energy, instrumentation** and **lyrics**. These factors only examine the relationship between the seed song and the playlists. Other factors (e.g., usage, time) are assumed to be implicit in the choice of the seed song and are not tested in this work. The subject can select as many factors as they deem relevant or indicate that the factor was not relevant (this choice was not pre-selected) before continuing on to the next trial. Subjects can quit at any stage and their progress is saved.

4.2 The Music

The playlists are built from the authors' personal music library of over 12,000 relatively popular songs that span the most common genres of Western popular music, with very little music outside these genres. The genres include rock, alternative, punk, soft rock, classic rock, folk, pop, electronica, experimental, blues, jazz, soul and hip-hop. The 50 seed songs were chosen to represent these genres in proportions roughly equal to those observed in the library [5] . For each seed song, we pre-calculate a five-song playlist using each of the recommender systems described in Section 3.

We used 30-second song clips, beginning 30 seconds from the start of the song. 30 seconds is generally enough to give a good impression of a song (e.g., it is a standard length for previewing songs in online music stores) while being sufficiently short to make each trial manageable, since subjects are required to listen to 11 clips.

In half the trials, song and artist names for both the seed song and the playlist songs are *hidden* from subjects. This allows us to investigate the influence of the song and, in particular, artist names on subjects' evaluation of playlists. In certain music recommendation scenarios, listeners may read the names of the songs they hear (e.g., album tracklists, record stores, music players such as iTunes, WinAmp,

[5] The list of seed songs and music library can be found at `http://cosmal.ucsd.edu/cal/projects/playlist/`

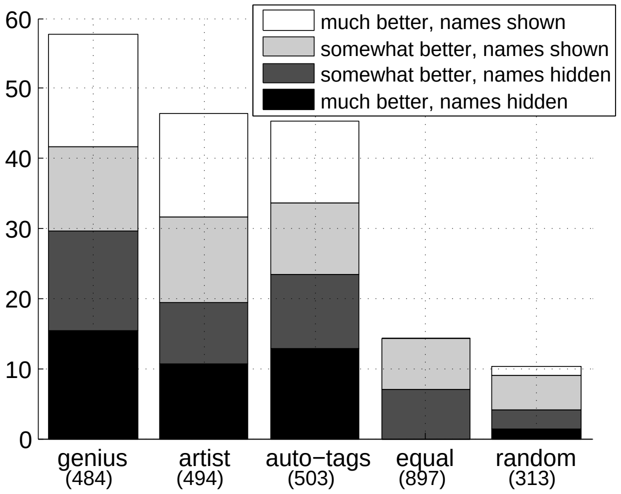

Figure 1. Percent wins for each music recommender system, divided over trials where the song names were hidden or shown. X-axis displays the system (and number of trials where this system was presented). Y-axis displays the percentage of trials where the system was the winner.

and last.fm) while in other cases, they only hear - and do not see - the playlist (e.g., most stereos, iPods on shuffle mode, on the radio, at parties or clubs).

5. RESULTS

Experimental subjects were recruited from psychology and engineering classes at UCSD, via email to friends, colleagues and the Music-IR mailing list and from online blogs and social networks. During the three week experiment, 185 subjects completed 894 trials with, on average, 4.8 trials per subject, including a maximum of 44 trials by one user. Seed songs were chosen randomly (without repeating a seed for any subject) and each of the 50 seed songs was presented an average of 18 times with a minimum of 11 and a maximum of 30 trials. Each of the three playlist generation methods was presented in at least 638 trials and each of the 150 playlists (a {seed song, playlist method} pair) was presented in, on average, 11.8 trials.

Figure 1 displays how often each recommender system won, as a proportion of all the trials in which it appeared. Figure 2 indicates how each system fared against the others, in head-to-head comparisons. The fading between colors in Figures 2-4 indicates the variance over 50 random subsamplings of 75% of the data for each condition. It is clear that Genius outperforms both the Artist Similarity and Similar Tags methods in most cases although a more detailed examination is given below.

5.1 Trial Lengths

Table 1 demonstrates that subjects spent, on average, 226 seconds on each trial, indicating that they listened to almost all of each 30-second song clip (11 songs x 30 seconds = 330 seconds). This time was significantly less for trials where the song and artist names were visible (196 seconds) and significantly longer when the names were hidden (258 seconds), indicating that subjects were often able to evaluate playlists (or, at least, some of the songs in a playlist) simply by looking at the names of the song and artist. Thus, we expect trials where the names were hidden to estimate better the impact of the "sounds" of the songs while those with names shown will demonstrate the impact of artist similarity.

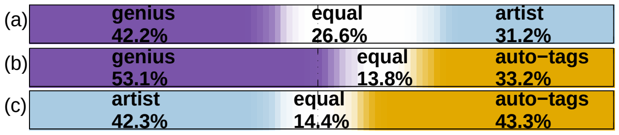

Figure 2. Head-to-head playlist comparisons over all conditions. Ignoring the equal votes, all systems are significantly better than random and Genius is significantly better than the content-based system using similar tags (Chi-square test for fit to a uniform distribution, $\alpha = 0.05$). All other differences are not significant.

Trial Length (sec)	Mean	Median
All Trials	226	150
Names Shown	258	165
Names Hidden	196	139

Table 1. Average seconds spent per trial as well as for trials where the song and artist names were shown or hidden.

5.2 Knowing the Names

Visibility of song and artist names had a large influence on how subjects evaluated each playlist. Showing the names benefited the metadata-based systems where, as evidenced by the shorter time spent on these trials, subjects made use of this metadata information to make their evaluations. Comparisons between each pair of algorithms are summarized in Figure 3. Of particular note is the comparison between the two metadata-based systems. When the names are shown, we see in Figure 3(a) that subjects actually rate the Artist Similarity playlists slightly better than the Genius playlists. This may indicate that social or visual cues are, at times, more salient than acoustic similarity or that, given some "explanation" of how a playlist is built, listeners are more forgiving of acoustic mismatches [3]. However, when the names are hidden, and subjects must base their judgements on the acoustics alone, Genius is overwhelmingly superior (Figure 3(b)).

5.3 Familiar and Liked Songs

The effects of subjects' familiarity with the seed song is shown in Figure 5. The effect of affinity for the seed song was qualitatively almost identical and is not shown. In both cases, Genius benefits from decreased familiarity or affinity while the Artist Similarity method suffers. In other words, when subjects did not know (or like) a song, and presumably could make less use of artist associations, they preferred Genius' recommendations. This is a strong indication that Genius does not just average over artists but determines song-specific similarities. The only statistically-significant

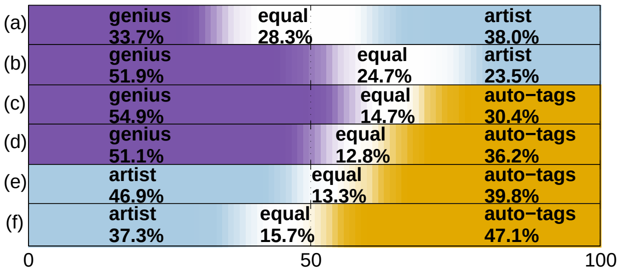

Figure 3. Head-to-head playlist comparisons over the condition where **song and artist names are shown** (a),(c)&(e) **or hidden** (b),(d)&(f). When names are shown (a), Artist Similarity outperforms Genius, but suffers significantly when names are hidden (b) (Chi-square test for independence, $\alpha = 0.05$). The content-based system always benefits when names are hidden (d),(f), forcing subjects to consider acoustics.

change between these conditions (familiar / unfamiliar or liked / not liked) is the reversal in ratings for the Artist and Tag Similarity methods. When familiar with the seed song, subjects were able to appreciate similar artists in the Artist Similarity playlists but, in the absence of this prior knowledge, acoustic similarity prevailed.

5.4 Content Similarity from Collaborative Filtering

We have seen that collaborative filtering finds similarities between songs, not just artists. Can collaborative filtering based on usage and purchase metadata actually capture similarity in acoustic content? To answer this question, we consider trials where subjects were *unfamiliar* with the seed song and where the names of the songs and artists were *hidden*. This removes the influence of song familiarity and artist associations so that subjects' evaluations are based only on acoustic similarity. We also required that subjects *like* the seed song so that they had sufficient motivation and experience with the genre to make relevant evaluations (many subjects reported that they felt unwilling or unable to evaluate songs they disliked). The outcome is shown in Figure 4 where it can be seen that Genius now performs at the same level as the system based solely on acoustic content. This agrees with the findings of Baccigalupo et al.[16] who provide evidence that information about song associations discovered from social playlists can be used to derive genre affinities i.e., collaborative filtering data can be used to derive aspects of acoustic similarity.

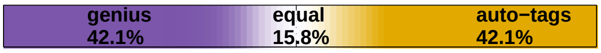

Figure 4. Genius captures content. When subjects were unfamiliar with a seed song that they liked and had no information about song and artist names (26 trials), Genius matches the performance of the content-based system.

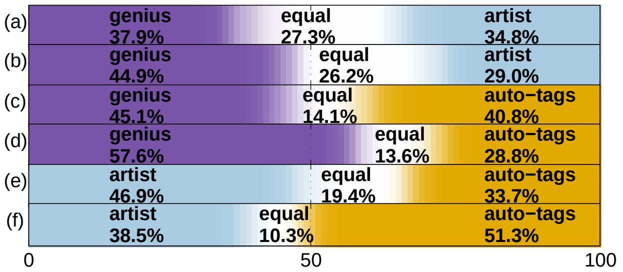

Figure 5. Head-to-head playlist comparisons over the condition where subjects are **familiar** (a),(c),&(e) **or unfamiliar** (b),(d)&(f) **with the seed song**. There is a significant difference between (e) and (f) where the content-based Tag Similarity system is more effective than Artist Similiarity when the seed song is not familiar. (Chi-square test for independence, $\alpha = 0.05$).

5.5 Bad Songs

Table 2 examines the "bad songs" in each playlist that subjects felt did not fit well with the seed song. Overall, the playlist with fewer bad songs won in 81% of trials.

Genius	1.30
Similar Artist	1.18
Similar Tags	1.33
Random	2.56

Trial Winner	1.20
Trial Loser	1.80

Table 2. Average "bad songs" in playlists from each system as well as the average for the winner and loser of each trial.

6. SMARTER THAN GENIUS

While Genius performs as well or better than the metadata- and content-based systems on our test collection of popular music, it is unable to make recommendations from the large "long tail" of new, undiscovered music. We now consider how a music recommender system could take advantage of both content-based information and metadata, when available, to perform as well or better than Genius, without the need for massive amounts of user data.

6.1 Balancing Content and Metadata

Table 3 quantifies the competing influences of artist and acoustic similarity. We show the average artist similarity and auto-tag KL divergence between the seed songs and all the songs from playlists generated by each recommender system. These measures are also shown for all the bad songs. By design, the content-based system has minimum KL and, although it can only access artist information indirectly through acoustics, it captures artist similarity at a better-than-random level. Though they produce good recommendations, both Genius and the Artist Similarity systems have significantly higher KL. This indicates that simply minimizing divergence between semantic descriptions will not produce perfect recommendations. Likewise, recommending similar artists is not sufficient as many bad songs had high artist similarity. Table 2 indicates that a recommender should avoid bad songs with very large semantic

	Artist Similarity	Tag KL Divergence
Genius	19.8	0.81
Similar Artists	**44.5**	0.89
Similar Tags	5.0	**0.14**
Random	1.1	1.15
Bad Songs	16.9	1.20

Table 3. Average artist similarity (between 0 and 100) and auto-tag KL divergence (larger means less similar) between a seed song and playlist songs recommended by each system as well as for bad songs produced by all systems.

differences (high KL divergence) while also making sure to include some clearly similar artists. For example, in 14 of the 50 playlists tested, Genius recommended a song by the *same* artist as the seed song, a simple way to enhance perception of the relevance of the recommendations.

6.2 Musical factors influencing playlist evaluations

Stage 3 of our experiment asked subjects to indicate how well the playlist songs matched the seed song on six different musical cues: similar style (genre), sound, artist, energy, instruments and lyrics. Subjects could indicate that a particular factor was most relevant to either playlist, even the one they had deemed inferior in stage 2. Figure 6 displays the percentage of trials where each system best manifested these factors. Genius playlists often match the styles (47%) and sounds (53%) of the seed song while, predictably, the content-based Similar Tags system rarely returns similar artists (26%). The percentages below the x-axis in Figure 6 indicate how often each factor was cited as a favorable influence (subjects were not required to mark these influences). Similarity between the sound of the seed and the playlist was the most frequently cited factor (82%) while similar lyrics rarely influenced playlist evaluation (36%).

7. CONCLUSIONS

We find that Genius' collaborative filtering approach, which essentially captures the wisdom of the crowds, performs well on a test collection of popular music. By removing evaluator bias resulting from artist names and song familiarity, we show that Genius captures song-specific aspects of acoustic similarity, as can be derived from a purely content-based system. Thus, for exploring the long tail, a content-based recommender can be expected to perform similarly to Genius, *if* collaborative filtering data were available.

We discover that seeing song and artist names has a significant effect on how a playlist is evaluated, indicating that recommender systems must be designed with applications in mind. We highlight the most influential factors on similarity evaluation and suggest that balancing content analysis to avoid bad songs with metadata similarity to provide transparent recommendations can help build smarter music recommender systems.

8. ACKNOWLEDGEMENTS

We thank Sarah Creel and Brian McFee for their advice. GL is supported by NSF grant DMS-MSPA 062540922.

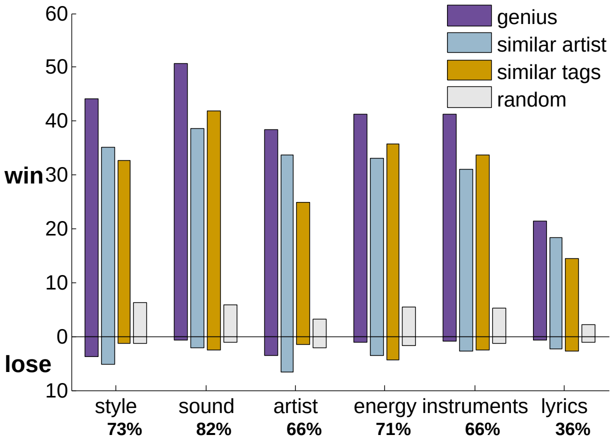

Figure 6. Musical factors influence song similarity. Y-axis shows how often each recommender system best matched musical factors of the seed song, averaged over trials evaluating that system (win or lose). Below the x-axis is the percentage of total trials where each factor was an influence.

9. REFERENCES

[1] S. Cunningham, D. Bainbridge, and A. Falconer. More of an art than a science: Supporting the creation of playlists and mixes. In *ISMIR*, 2006.

[2] M. Salganik, P. Dodds, and D. Watts. Experimental study of inequality and unpredictability in an artificial cultrural market. *Science*, 311(5762):854–856, 2006.

[3] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *RecSys*, 2008.

[4] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR*, 2002.

[5] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *ISMIR*, 2008.

[6] L. Xiao, L. Liu, F. Seide, and J. Zhou. Learning a music similarity measure on automatic annotations with application to playlist generation. In *ICASSP*, 2009.

[7] B. Fields, C. Rhodes, and M. Casey. Social playlists and bottleneck measurements: Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *ISMIR*, 2008.

[8] F. Vignoli and S. Pauws. A music retrieval system based on user-driven similarity and its evaluation. In *ISMIR*, 2005.

[9] L. Barrington, A.B. Chan, D. Turnbull, and G. Lanckriet. Audio information retrieval using semantic similarity. In *ICASSP*, 2007.

[10] Gracenote Inc. Automatic identification of sound recordings. US Patent Number 7,328,153, 2008.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 16(2):467–476, February 2008.

[12] N. Rasiwasia, N. Vasconcelos, and P. Moreno. Bridging the gap: Query by semantic example. *IEEE Trans. on Multimedia*, 2007.

[13] J.S.Downie. Music Information Retrieval eXchange. *ISMIR*, 2007.

[14] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *ISMIR*, 2005.

[15] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evalutation of acoustic and subjective music-similarity measures. *Computer Music Journal*, pages 63–76, 2004.

[16] C. Baccigalupo, E. Plaza, and J. Donaldson. Uncovering affinity of artists to multiple genres from social behavior data. In *ISMIR*, 2008.

[17] E. Law and L vonAhn. Input-agreement: A new mechanism for collecting data using human computation games. In *ACM CHI*, 2009.