Presented at ISMIR 2007, Vienna, Austria, September 2007.

SIMILARITY BASED ON RATING DATA

Malcolm Slaney
Yahoo! Research
2821 Mission College Blvd.
Santa Clara, CA 95054
malcolm@ieee.org

William White
Yahoo! Media Innovation
1950 University Ave.
Berkeley, CA 94704
wwhite@yahoo-inc.com

ABSTRACT

This paper describes an algorithm to measure the similarity of two multimedia objects, such as songs or movies, using users' preferences. Much of the previous work on query-by-example (QBE) or music similarity uses detailed analysis of the object's content. This is difficult and it is often impossible to capture how consumers react to the music. We argue that a large collection of user's preferences is more accurate, at least in comparison to our benchmark system, at finding similar songs. We describe an algorithm based the song's rating data, and show how this approach works by measuring its performance using an objective metric based on whether the same artist performed both songs. Our similarity results are based on 1.5 million musical judgments by 380,000 users. We test our system by generating playlists using a content-based system, our rating-based system, and a random list of songs. Music listeners greatly preferred the ratings-based playlists over the content-based and random playlists.

1 INTRODUCTION

This paper describes a means to evaluate the similarity of two multimedia objects using users' stated preference or rating data about items in the collection. We use a large music collection to illustrate this work, but the same idea applies to any collection of objects where many users report their preference of an object.

Most work on similarity uses content-based algorithms. A specialized algorithm looks at the content (usually music) calculates various musically-inspired measures of the sound to form a feature vector, and then compares two features vectors to make a decision about similarity [1]. This similarity measurement is the heart of conventional query-by-example (QBE) systems, such as QBIC [2]. But, even measuring similarity with human raters is difficult [4].

Our work ignores the content. Instead we look at a large group of users and ask if these users, with a wide range of personal musical interests, rate two pieces of music in the same manner. If jazz, classical, blues, and hip-hop lovers all rate two songs in the same way, whether they hate it or love it, then the songs are most certainly similar.

This paper hypothesizes that ratings data from a large number of users, averaging over many different kinds of tastes, produces an accurate measure of similarity. We compare our approach to a traditional content-based approach. We test this approach by generating a list of songs which are most similar to the query. We show the superiority of our approach by asking for human judgments and by counting songs identified as similar by virtue that they are performed by the same artist (a weak, but highly objective measure of similarity, as we will discuss in Section 5.3.)

2 MOTIVATION

There are many situations where it is useful to know which media is similar to other media. We would like to be able to find songs similar to a user's interests, or to create a playlist that sequences a list of songs in a pleasing manner. We are interested in analyzing all types of media.

With the emergence of digital media, users are now faced with the problem of too much choice. There is simply more media available to the user now than they will ever be able to consume. Traditional media-discovery methods such as searching for a track by artist, album or title are no longer sufficient. The user is restricted to what they already know, which is a constantly shrinking piece of an ever increasing media pie.

Using genre as the primary discovery mechanism is also not sufficient, as genre is often difficult to pin down, with two people often classifying the same piece of music into completely different genres. Clearly, there is an urgent need for new methods, which allow users to discover new media that is fresh and exciting to them.

Personalized recommendation systems are one of the more popular mechanisms for facilitating media discovery. However, most recommendation systems need to get a fair amount of information about the user and what she likes before they become effective. An advantage of our approach is that it does not require any user-specific ratings to be effective. Our method can start with a single seed song, without the need to build an elaborate user profile.

Generating a list of music based upon their similarity

to a specific track facilitates a more interactive musical experience. By creating similarity-based playlists, we can provide a highly targeted experience, appropriate for the situation, engaging for the user, and easy for them to initiate.

One can imagine using such a technology to generate the perfect on-demand mood music experience. You wake up on a lazy Sunday morning and start things off with "Girl From Ipanema" by Stan Getz. Similar music would follow for a couple hours, easing you into the day. Then after lunch, you make a single adjustment—changing the seed track to "We Will Rock You" by Queen. The mood picks up and the tracks that follow make up the perfect soundtrack for watching Sunday afternoon football over beers with friends. By focusing on similarity to a given source track, we can deliver music that the user not only generally likes, but will want to hear right now.

In practice, one would combine this approach with a conventional content-based analysis approach (like our previous work on genre-gram based similarity [9], or the best of the MIREX work on similarity search [1]) since we do not have rating data for everything.

Our approach is different from collaborative filtering, which uses rating data to find songs that a user likes [6]. Collaborative filtering is usually structured as a rating-prediction problem. Given a collection of users' ratings, collaborative filtering builds a model of the song-rating database. This allows one to predict a new user's ratings on any new song based on what similar users like. This might be done using nearest-neighbor approaches [8], or by clustering users and then examining the average rating in the cluster [3].

An Internet radio service called Pandora uses a hand-labeled meta-data feature, i.e. "Angry Lyrics" and "Backbeat Hand Claps," to find similar songs [11]. Then based on user feedback, they adjust the relative weights of the different parameters to find music for each listener. This is different from our work since our underlying feature vector is based on musical preferences.

Our work is similar to Whitman's work [12], which uses text from the web to judge artist similarity. They use rich descriptions about the artist and their work with hundreds of words written by a small number of web authors. We, on the other hand, use less than 3 bits per user (usually one point on a five-star scale), a relatively inexpensive amount of data to collect, because we believe similarity is better judged by more listeners.

3 RATINGS-BASED SIMILARITY APPROACH

Consider three listeners: U_{jazz}, U_{rock}, $U_{classical}$. They might rate three songs as follows on a five-point scale (0 means they hate it, 5 means they love it) as shown in Table 1.

From this small snippet of data, we can infer that songs s_1 and s_3 are similar because jazz and rock lovers like them and the classical listener does not. (We can not say anything about song s_2.)

	s_1	s_2	s_3
U_{jazz}	5	0	5
U_{rock}	5	0	5
$U_{classical}$	0	5	0

Table 1. A sample table of user ratings for three songs and three users.

The algorithm we describe below quantifies these similarities and differences among users.

4 ALGORITHMS

Our approach starts with data about users' preferences for different media (i.e. songs or films). The variable $r(u, s)$ describes user u's rating for object s. The vector $\vec{r_s}$ is an N_u-dimensional vector of all the rating data for object s. These ratings range from 0, which means never play this again, to 100, which means the user "can't get enough of it." In a large database of users and media, many ratings will be undefined.

To compute the similarity of two objects, we compute the normalized vector

$$\vec{r_s}' = (\vec{r_s} - b)/|\vec{r_s} - b| \quad (1)$$

where b is a bias value that will eventually represent the implicit rating for an unrated object. This normalization is important because it allows songs with different numbers of raters to be compared. A ratings database of this size is necessarily sparse—almost nobody will rate all songs. We discuss the effect this has on our system in Section 5.4. By default, we give unlabeled songs in our system an implicit rating of b.

The similarity of two objects, s_1 and s_2 is written

$$s = \vec{r_{s1}}' \cdot \vec{r_{s2}}'. \quad (2)$$

This is equivalent to the cosine metric used when comparing the word-frequencies in conventional text-based information retrieval. It is also monotonically related to the Euclidean distance between the two vectors since

$$|a - b|^2 = |a|^2 - |2ab| + |b|^2 = 2 - |2ab| \quad (3)$$

and the last step follows since the magnitude of our ratings vectors is equal to 1.

Given a query we generate a list of similar songs, or a playlist, by finding the songs that have the highest similarity, as judged by Equation 2.

We compare the ratings-based similarity to a content-based scheme using a genre-gram [10]. In the original work on genre-grams, an hand-crafted collection of features is combined to form a feature vector. Different positions in this vector space indicate different musical genres. We extended this idea in our previous work [9] by using a multi-dimensional linear discriminant analysis (LDA) to rotate the coordinate axis and generate an optimal 28-dimensional subspace. Similarity in genre space is defined as the Euclidean distance between the positions of

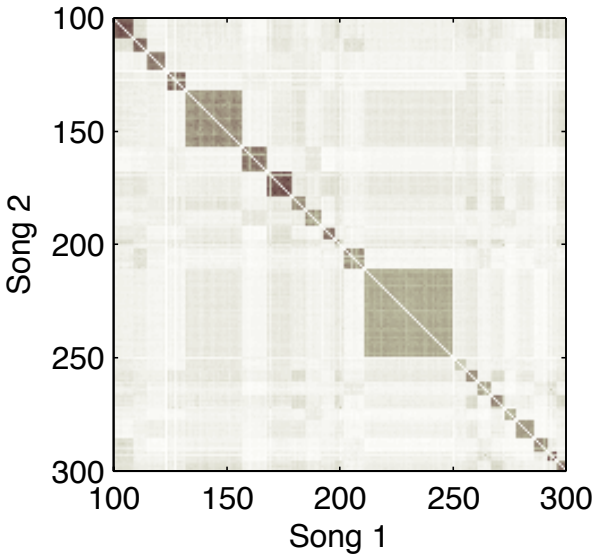

Figure 1. A portion of the similarity matrix for 200 of the songs in our database.

two points in genre space. In this work we used Euclidean distance in feature space since the results were similar to the LDA result.

5 RESULTS

5.1 Database

We tested the ratings-based similarity algorithm using a database of 1,449,335 ratings of jazz songs provided by users of the Yahoo! Music service and described by Marlin [5]. From this list of jazz songs we choose the 1000 songs with the most ratings. The number of ratings per song ranged from "Sunrise" by Norah Jones with 98,658 ratings to a song by Os Nosos with 118 ratings. 380,911 users contributed to these ratings, with one user rating 913 songs and many users rating only one of these jazz songs.

Thus we restricted our initial study to music which fell under the category "Jazz," and then restricted it even more to those tunes for which we had ratings data. Essentially this gave us a set of tunes which are not specific enough to be one specific type of jazz (i.e. bebop or fusion), but jazzy enough to be labeled "Jazz." This explains why Nora Jones, a "contemporary pop" artist, got top ranking in our results.

Figure 1 shows a portion of the similarity matrix for our 1,000 song database. Pixels representing two songs that are most similar show up darker in this matrix. (The diagonal is set to zero so we can more easily see the other songs.) The squares along the diagonal represent groups of songs, that happen to be listed in order in our database, from the same artist or album. These songs tend to be highly similar. (Note: The Yahoo! Music system allows users to rate artists too, but we did not use this data in our analysis, only the per-song rating data.)

Rank	Song Title	Artist
Query	Those Sweet Words	Norah Jones
1	Shoot The Moon	Norah Jones
2	What Am I To You?	Norah Jones
3	Sunrise	Norah Jones
4	Seven Years	Norah Jones

Table 2. Rating-based similarity: Most similar tracks for the query song "Those Sweet Words." The 4 most similar songs are all recorded by Norah Jones.

Rank	Song Title	Artist
Query	Those Sweet Words	Norah Jones
5	Come Live You Life With Me	Peter Cincotti
6	Dansons La Gigue	Patricia Barber
7	Noa Noa	Wolfgang Dauner
8	Ain't Nobody Here But Us Chickens	Louis Jordan

Table 3. Rating-based similarity: Most similar tracks for the query "Those Sweet Words," as in Table 2 but not including songs by Norah Jones.

5.2 Sample Playlists

Tables 2, 3, and 4 shows results for both algorithms for the query song "Those Sweet Words" by Norah Jones. In Table 2, we demonstrate our ratings-based measure and show the top 5 songs. Then in Table 3 we remove all songs by Norah Jones and show the next four selections. Finally, in Table 4, we show the results from a content-based approach. In all cases, we can only measure similarity when we have enough ratings data, the 1,000-song subset of jazz songs in our database.

5.3 Artist-based Similarity Test

For the purposes of an artist-based test, we say songs are similar (a binary decision) if they are recorded by the same artist. We use the artist of a song as a weak, but objective measure of two songs similarity. This metric is potentially weak for two reasons. First, an artist's career can span

Rank	Song Title	Artist
Query	Those Sweet Words	Norah Jones
1	More	Harry Connick, Jr.
2	I Really Love You	Cy Coleman
3	Sway	Peter Cincotti
4	I Wants To Stay Here	Ella Fitzgerald
5	Feelin' The Same Way	Norah Jones

Table 4. Content-based similarity: Most similar tracks for the query "Those Sweet Words" by Nora Jones.

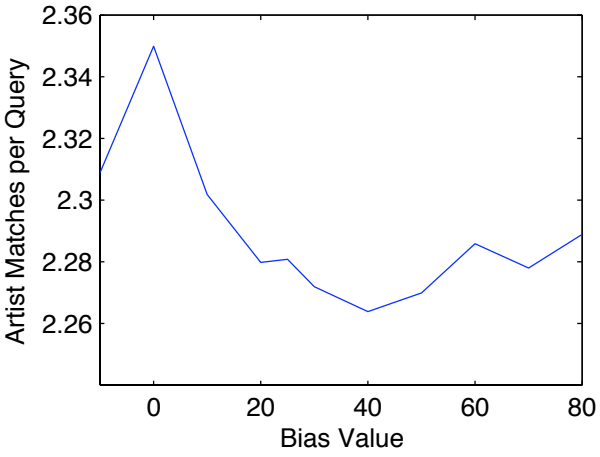

Figure 2. Average number of artist matches per query for the 50 most similar songs—computed as a function of the normalization bias, or the implicit rating of unlabeled data. Higher numbers are better because they indicate a greater agreement between the ratings-based similarity measure and the artist who recorded the best matches.

many decades and musical styles. Second, and harder to test, our data is not complete and the data that is missing is not random (see Section 5.5). It is possible that users might rate all songs in an album, or their ratings might reflect their interest in the artist and not the song. Yet in spite of these two flaws, the metric is easy to compute. It allows us to study parameter variations and test the entire collection of song-similarity judgments, which we can not do with human listeners.

5.4 Implicit Rating

The dot-product formalism we defined in Equation 2 does not define what happens when data is missing. A common solution in collaborative filtering evaluates the dot product only over those users who rate both songs. Since Equation 2 computes a sum over all users, dropping data is equivalent to multiplying by 0, or assuming all the missing data is equal to 0. This means that the bias term in Equation 1 has special significance as the implicit rating for a song.

We measured the effect of changing the bias value when normalizing the rating data by measure the same-artist rate for the genereated playlist. The results in Figure 2 show that a bias of 0 is the best.

At the start of our work, we hypothesized that a bias of 50 might be most effective. We rationalized that positive and negative results might be equally important for measuring similarity. This proves not to be true, perhaps because the rating data is not uniform [5], or because people who like a song are the best people to evaluate similarity.

5.5 Missing Data

A histogram of all data in our entire music-rating database, Figure 3, shows peaks for the lowest and highest ratings. Yet a histogram of rating data when a small number of

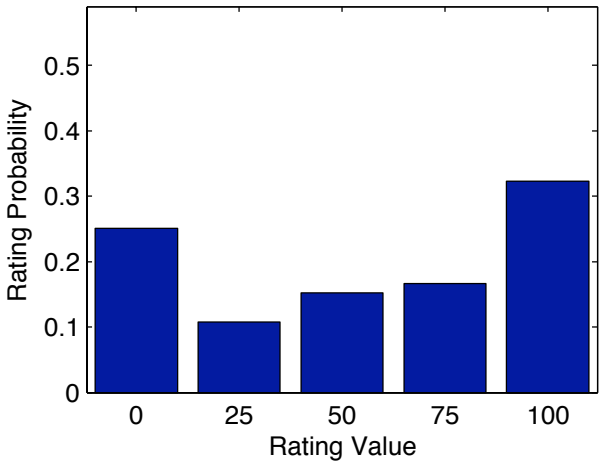

Figure 3. Probability of rating over our entire database (717 million ratings of 136,000 songs given by 1.8 million users of Yahoo! Music services and collected between 2002 and 2006). (Adapted with permission from Marlin [5].)

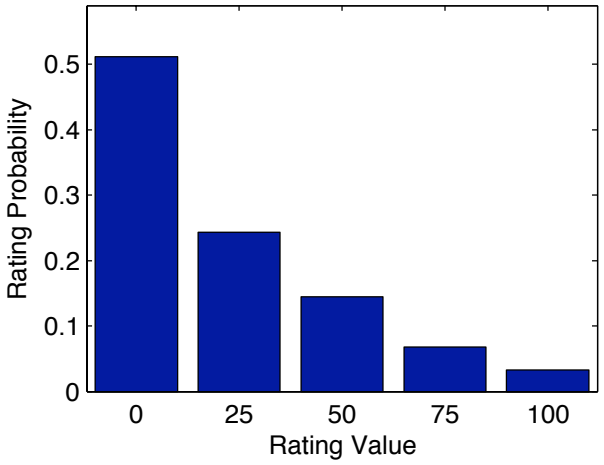

Figure 4. Probability of rating for 35,786 users who each rated 10 songs that we chose at random. (Adapted with permission from Marlin [5].)

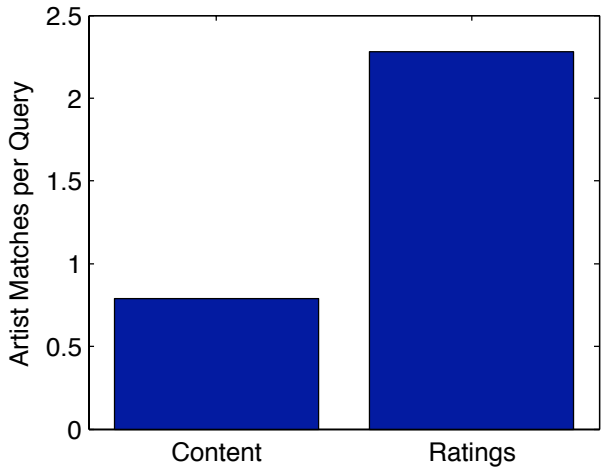

Figure 5. Average number of artist matches per query for the 50 most similar songs for a content-based algorithm on the left and our rating-based approach on the right. Higher numbers are better because they indicate a greater agreement between the ratings-based similarity measure and the artist that recorded the best matches.

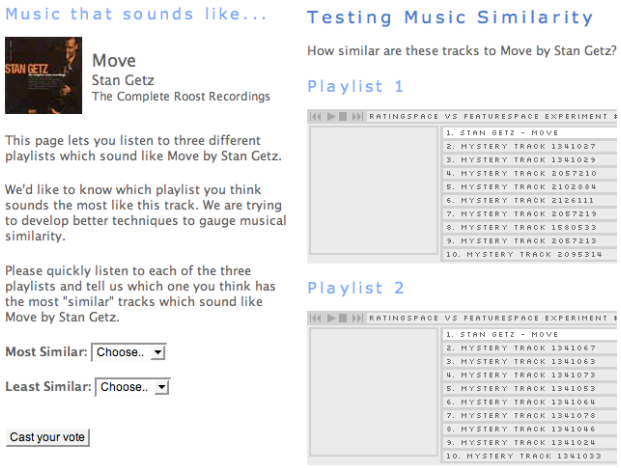

Figure 6. A portion of the screen showing two of the three playlists in our user test.

users are asked to rate all songs in a sample set, Figure 4, shows a different pattern, with users disliking most songs and only giving their highest ratings to a select few songs.

The difference between Figures 3 and 4 is that the Figure 3 shows the ratings of songs that users hear (either by their choice, based on a search, or recommended to them by the service) *and* they choose to rate. This is significant because users tend to really hate or really like the songs they hear. The latter figure is a measure of what users think of *all* songs in the database. For this figure, a small number of volunteers rated a completely at-random set of songs, exposing them to songs to which they would normally not listen to.

For whatever reason the best default rating is close to zero. Perhaps it is because they do not rate songs that they do not care about or because in practice users are not exposed to songs that they will not like.

5.6 Artist-Based Similarity Test

Figure 5 shows a test comparing the content-based and ratings-based similarity measures. In this test we judged which algorithm produced a better, or more similar playlist, by asking which list had more songs by the artist that performed the query song. The rating-based similarity measure described in this paper works almost twice as well as the content-based scheme we use as a benchmark. One can argue that this content-based scheme might not be as good as others in the literature, but this certainly shows that the rating-based scheme is competitive, and we believe this approach will only get better with more data.

Approach	Most Similar Votes	Least Similar Votes
Random	1	13
Content	1	4
Rating	16	1

Table 5. Results of a user test comparing three different means of calculating similar songs. Users were asked to vote for the playlist with the songs most similar to the query, and to vote for the playlist with the songs least similar.

5.7 User Test

Finally we also performed a small user test to confirm our analytic results. Users were shown a display with a single query song, and three lists of songs: similar songs based on content, similar songs based on ratings, and a random list of songs (Figure 6). The songs were in order of their similarity, but were not identified in any manner. The order of the three lists was randomized, and each list showed the 10 most similar songs based on the query. For each song there is only a button that the user can press to hear a 30-second snippet of audio—other than the query song, the songs were not identified.

Table 5 shows our performance with human judgments. By a wide margin, the 18 listeners judged the playlist generated by the rating-based approach to produce the most similar list of songs, and the random list to be least similar. This is true whether the listener liked or disliked jazz. This result is important because it shows that a little bit of data from a large number of people expressing their preference for music can be used to measure a completely different question—are two songs musically similar?

6 CONCLUSIONS

Similarity is both difficult to measure and highly personal. A classical-music lover will recognize the underlying similarity of a Bach recording on period pieces and Wendy

Carlos' "Switched on Bach" recordings; but he will probably love one and hate the other—they are not very similar in his mind. A teenage lover of hip-hop will instantly hate both renditions. Perhaps this explains why positive ratings are more important for similarity judgments than negative—lovers of a particular style of music are more discriminating. We believe that averaging the musical interests of many users—380,911 in this study—is the best answer to the "what is similar" problem.

We have described a method that computes the similarity of two songs by comparing users' ratings or preferences for the two different pieces of music. We used a large collection of rating data to judge the similarity of 1000 jazz songs. Using both a simple, but objective measure of similarity, whether the songs were recorded by the same artist, and a small user test we demonstrated the superior performance of the rating data over an approach based on analysis of the content. Content-based methods will always be necessary for songs that are new, or are not popular enough to warrant a large number of listeners. The rating-based approach can be applied to any data—music, books, films—where we have rating data.

7 ACKNOWLEDGMENTS

We appreciate the help we have received from Mike Mull and Ben Marlin who provided the data, and helped us to analyze and understand this data. We appreciate the assistance of Sara Anderson and the anonymous reviewers in improving the presentation of this work.

8 REFERENCES

[1] J. Stephen Downie. The Music Information Retrieval Evaluation eXchange (MIREX). In *D-Lib Magazine,* 12 (Issue 12), 2006:

[2] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, Peter Yanker. Query by image and video content: The QBIC System. *Computer*, 28(9), pp. 21–32, 1995.

[3] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. in *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, pp. 688–693, 1999.

[4] M. Cameron Jones, J. Stephen Downie, Andreas F. Ehmann. Human similarity judgments: Implications for the design of formal evaluations. *Proceedings of the 2007 International Society of Music Information Retrieval,* Vienna, 2007.

[5] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. Collaborative filtering and the missing at random assumption. To be published in *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence.* 2007.

[6] David M. Pennock, Eric Horvitz, Steve Lawrence, and C. Lee Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach, in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000),* pp. 473–480, Stanford, CA, June 2000.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An open architecture for collaborative filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work,* pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[8] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th International World Wide Web Conference (WWW10),* Hong Kong, May 2001.

[9] Malcolm Slaney and William White. Measuring playlist diversity for recommendation systems. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, Santa Barbara, California, USA, 27 October 2006.

[10] George Tzanetakis, Georg Essl and Perry Cook. Automatic musical genre classification of audio signals. In *Proceedings International Symposium for Audio Information Retrieval (ISMIR)*, October, 2001.

[11] Tim Westergreen. Pandora Town Hall, CCRMA, Stanford University, 29 November 2006.

[12] Brian Whitman and Steve Lawrence. Inferring descriptions and similarity for music from community metadata. In *Voices of Nature, Proceedings of the 2002 International Computer Music Conference.* pp 591–598. 16–21 September 2002.