Measuring Playlist Diversity for Recommendation Systems

Malcolm Slaney
Yahoo! Research Labs
701 North First Street
Sunnyvale, CA 94089
malcolm@ieee.org

William White
Yahoo! Music
2700 Pennsylvania Avenue
Santa Monica, CA 90404
wwhite@yahoo-inc.com

Abstract
We describe a way to measure the diversity of consumer's musical interests and characterize this diversity using published musical playlists. For each song in the playlist we calculate a set of features, which were optimized for genre recognition, and represent the song as a single point in a multidimensional genre-space. Given the points for a set of songs, we fit an ellipsoid to the data, and then describe the diversity of the playlist by calculating the volume of the enclosing ellipsoid. We compare 887 different playlists, representing nearly 29,000 distinct songs, to collections of different genres and to the size of our entire database. Playlists tend to be less diverse than a genre, and, by our measure, about 5 orders of magnitude smaller than the entire song set. These characteristics are important for recommendation systems, which want to present users with a set of recommendations tuned to each user's diversity.

Categories and Subject Descriptors
H.5.5 Sound and Music Computing

General Terms
Algorithms, Measurement, Human Factors

Keywords: diversity, recommendation system, song similarity.

1. Introduction
Consumers now have access to an unprecedented amount of media. In particular, music databases allow users to choose from millions of songs, all available at the click of a mouse. For this reason, recommendation systems have become an important way for people to find new music. A new user's rating data over a small set of songs is combined with ratings data from a large number of other users to predict how the new listener will react to the rest of the catalog. Performance is often measured by the mean prediction error.

AMCMM'06, October 27, 2006, Santa Barbara, California, USA.

It is easy for a recommendation system, often implemented as a collaborative filtering system [4], to say which song has the highest rating, but these systems do not say anything about the range of songs a particular user might want to listen to. Users do not want to listen to the highest rated song over and over again. Instead there are various ad-hoc methods to broaden the playlist—increasing the diversity of the results, exposing the user to new music, and hopefully increasing customer satisfaction [7].

Characterizing diversity is one step in a complete recommendation system. Previous work describes approaches, for example, to bias recommendations to encourage choices in new directions [11]. Music is a more difficult problem than text retrieval because each user consumes dozens of recordings in a single sitting. Not only do we want to specify new directions but we want to know how far to go. This work gives us a way to measure a user's interests—one aspect of a recommendation system that includes diversity.

In this paper, we describe a method for measuring the diversity of a user's musical interests, and characterize 500 user's musical diversity. We know that different users have different musical interests. This work allows us to quantify their interests. This information will allow us to automatically generate better music playlists.

Music playlists allow us to measure the diversity of user's interests in ways that other e-commerce systems do not have. Users might buy just one book from a bookseller, the one at the top of their list; similar behavior is probably true for movie recommendations. But users will listen to music a number of times, and consume many more musical titles than they do other kinds of media. Thus it makes sense for us to study musical playlists, and understand how broad user's interests are.

It is difficult to measure the similarity (or differences) between two songs. Similarity is a personal decision and can depend on subtle semantic issues that are difficult to measure [5].

Instead, we use Tzanetakis' GenreGram to put a song into an acoustic space [8]. The GenreGram is used to describe musical style and was one of the early genre-recognition systems. One can argue that genre is often a meaningless marketing label, but nevertheless, the features used to decide genre can be useful for characterizing musical style. Tzanetakis' system defines a genre space using an assortment of acoustic features.

For our work we do not need a precise measure of similarity, just a way to reliably place songs in a musical space so that we can characterize a user's musical interests. User Gloria prefers music that has a strong beat. User Joshua wants soothing music for work. In both cases we want a measure of how broad their definition of strong beat or soothing music is.

We use the following approach in this paper. We collect a large number of playlists from the Internet and analyze each song with the features used to create a genre-gram. We further optimize our representation using linear-discriminant analysis (LDA) to find a low-dimensional linear subspace that best discriminates the different genres.

2. Related Work

Much of the work on diversity has been in the context of search results. One wants to return some results for all the different kinds of "jaguar" so that all users get useful results. Just returning automobile links or animal links might upset those looking for the other kind of result.

The theoretical justification for this result is based on minimizing the risk of not satisfying a user [10]. This means that the best result combines near-optimal results from a number of different facets. But this work does not characterize the breadth of a user's interests.

These ideas were implemented in a system that balances relevance and diversity [1]. Several different approaches are described, controlled by a knob that adjusts the tradeoff.

The MIREX competition has recently studied song similarity. One work [6] used a set of acoustic features, clustered the resulting vectors, and then computed a global song similarity. Our work uses a similar set of features, and trains an optimal set of features for the genre classifier, and then builds our diversity measure on top of these features.

3. Data Collection

We used the web playlist community, WebJay[1] as a source of user generated playlist data. WebJay enables users to build web playlists of audio tracks that are freely available on the Internet. The modern day equivalent of the mix tape, web playlists can be listened to via an RSS feed with a single click.

People are drawn to playlist sharing sites like Webjay to find new music and to share their own music taste with others. Capturing and contributing your "music personality" in the form of a playlist is a common theme amongst many "next generation portals" (eg. myspace.com) and Webjay users seem to take pride in their playlists, seeing them as a way to build an "online presence".

Playlist themes were diverse, ranging from analytical— "songs with super chromaticality," to political— "bush-loathing in music and song," functional— "music to skate to," to romantic— "Classic jazz vibes and others to go with pasta, spicy tomato-based sauce and red wine," comical— "tell Bill Clinton to go and inhale" to shameless, self-promotion, "ALL MY SONGS ARE PIMPIN!!!!! LEAVE ME A COMMENT, I WILL LEAVE ONE ON YOURS IF YOU DO MINE!!!!" These personal descriptions show that the authors see their playlists as important and representative of parts of their own personalities.

The 500 most popular WebJay playlist authors were found by crawling the popular playlists page and each playlist was downloaded as XSPF[2], parsed and added to a database. These playlists contained 86,130 track entries pointing to 58,415 unique web media URLs. We checked all 58,415 files and found 28,956 audio/mpeg tracks (or over 2500 hours of music) that we were able to download and analyze.

We used genre information about many of the songs to tune our feature set. The consistency of the genre metadata field, a free text field that can vary greatly depending upon the interface of the audio encoder being used, left something to be desired. Little more than 54% of the tracks we examined contained any genre metadata at all. Amongst these tracks, there were more than 950 different unique values populating the genre field. Based on the number of available songs, eleven of these genres were selected for classification purposes, spanning over 3500 tracks from our dataset. (See Figure 6 for a list of the genres we used.)

4. Data Analysis

Our primary goal is to measure the diversity of a set of songs. We perform this task by building a genre-recognition system, where each song is represented as a single point in a multidimensional acoustic feature space. We hypothesize that a musical space that allows us to easily discriminate different genres will also allow us to characterize song similarity. Given the points in space corresponding to each song in a playlist, we fit an ellipsoid to the data and calculate the volume of the set.

In this section we talk about the calculations we perform for creating the GenreGram, and then how we use these features to define the diversity of a set.

Our processing starts with MP3 files from a playlist. These files are converted, using FFMPEG[3], into 22kHz WAV files. We skipped the first 30 seconds of each song, and then extracted the next 30 seconds for audio analysis. These samples (over 240 hours of audio) were analyzed using MARSYAS to derive the genre-gram audio feature set [9].

Marsyas has a number of built-in algorithms for analyzing sound. The basic features used in this work operate over one or two frames of the sound and are:

[1] http://www.webjay.org

[2] http://www.xspf.org

[3] http://ffmpeg.sourceforge.net

- Spectral Centroid: The center-of-gravity of the magnitude spectrum—A measure of the brightness of the sound.
- Spectral Rolloff: The frequency in the magnitude spectogram for which 85% of the energy falls below. This is another measure of the timbre of the sound.
- Spectral Flux: The amount of change in the spectrum between frames. This is computed by squaring the difference between successive spectrogram frames.
- Zero Crossings: The number of sign changes in the acoustic waveform over a window. This is a measure of the dominant frequency in the signal.

For each of these four basic features, four different statistics are calculated. They are as follows:

- The mean of the mean: Calculate the mean over 40 frames, and then calculate the mean of this statistics. This is equivalent to a single calculation of the mean over the entire 30 seconds.
- The mean of the standard deviation: Calculate the standard deviation of the audio feature over 40 frames, and then calculate the mean these standard deviations over the entire 30 seconds. We want to know how the music changes over small windows of time.
- The standard deviation of the mean: Calculate the mean of the feature over 40 frames, and then calculate the standard deviation of the feature. The 40-frame window size gives us a reliable measure of the feature over a short window, and then we want to understand how it changes during the music.
- The standard deviation of the standard deviation: Calculate the standard deviation of the feature over 40 frames, and then calculate the standard deviation of this measure over the 30 seconds. This tells us how much change is there in this feature.

These four features and their four global measures give us 16 features. In addition there are 8 features that measure the rhythmic content of the music. The beat histogram is calculated by measuring the temporal correlation of the energy in the signal over windows of up to 1.5 seconds. The first two peaks are identified in this beat histogram and their properties are captured as features. The 8 rhythmic features are:

- High Peak Amplitude: the size of the biggest peak in the beat histogram.
- High Peak Beats-per-minute: the speed of the primary (or loudest) beat.
- Low Peak Amplitude: the size of the second-biggest peak in the peak histogram.
- Low Peak Beats-per-minute: the speed of the second-loudest beat.
- Peak Ratio: Ratio of the amplitude of the second peak to the amplitude of the first.
- Three features based on energy measures.

We perform a number of simple statistical transformations on the raw feature data before assigning the musical piece to a point in genre space.

First, we normalize each dimension by removing the mean and scaling so that its standard deviation is 1. This scaling, in particular, is necessary so we can perform the second step and get meaningful answers--at this point we know nothing about each dimension's value towards predicting genre space.

Second, we use the singular-value decomposition (SVD) to rearrange the dimensions to find the optimal low-dimensional approximation to each data point. The SVD has the property that the new dimensions (eigenvectors) are ordered so that the first N dimensions describe the input space with the lowest-possible error for any N-dimensional set of axis. This is important because we are interested in the best two-dimensional approximation so we can more easily visualize the genre space. In this work we use all 24 rotated dimensions as input to the decision stage.

Third, and finally, we use multi-class linear-discriminant analysis (LDA) to find the best set of orthogonal dimensions that allow us to clearly segregate the data into different classes [3]. In normal two-class LDA, a vector is returned that characterizes the hyperplane that best separates the labeled data. We do the same for the labeled genre data.

We characterized the different output representations by testing their performance in a genre-classification experiment. We chose seven of the medium-sized genres (between 100 and 900 songs per genre) and measured the genre-classification performance with cross validation. (We did this test 10 times, each time randomly selecting about 90% of the genre data as training examples, and then testing the performance of the classifier on unseen data.) In each case, with the number of LDA output dimensions between 1 and 24, we used a multi-class support-vector machine (SVM) to classify the testing data [2].

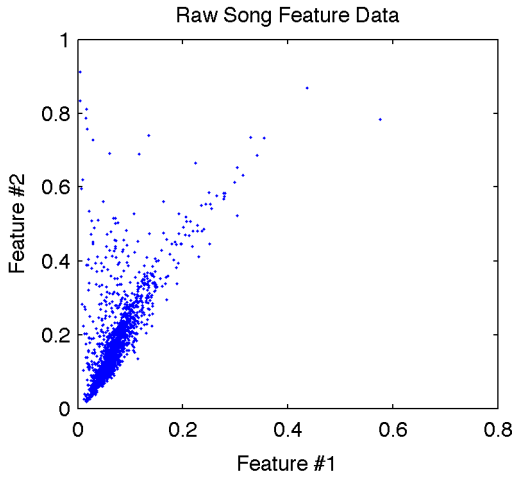

Figure 1. Raw feature data for two (arbitrary) dimensions.

Figures 1 through 4 show several plots that characterize the feature analysis stage. Figure 1 shows the raw feature data—we are plotting just two (arbitrary) acoustical-feature dimensions. All dimensions had similar scatter. Each point in the figure is the location of one song in this ultra-low-dimensional feature space.

Figure 2 shows the result after transforming the data into the best two-dimensional representation using a SVD. The eigenvalue analysis showed an exponential falloff, with no discernible breakpoint. Each musical piece, a point in this 2-D SVD space, is coded with a different color for each genre. There is still quiet a bit of overlap in the classes with a 2D projection.

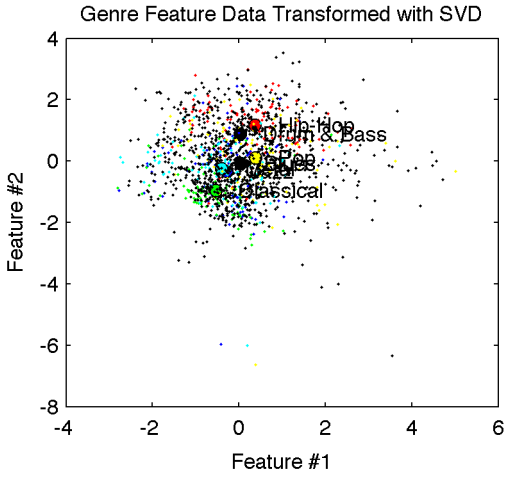

Figure 2. Scatter plot of the song data transformed using SVD. Only the first two dimensions are shown. Each color represents a different genre.

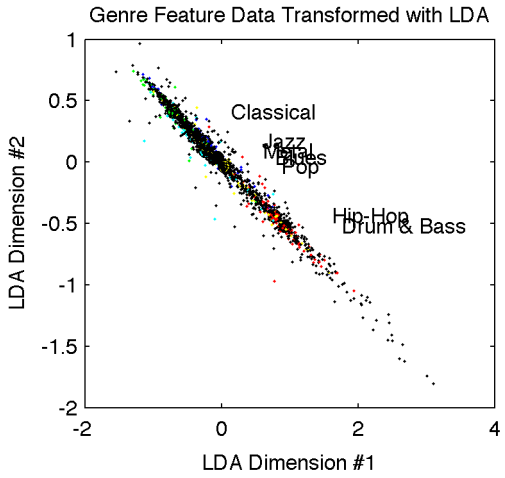

Figure 3. Scatter plot of each song's feature set after LDA transformation. Only the first two dimensions are shown.

Figure 3 shows the result after a 2-dimensional LDA analysis. Different genres are stretched along a line in this particular 2-dimensional subspace. (Other samples of the result of this LDA analysis were not so clear in the 2-dimensional projection.)

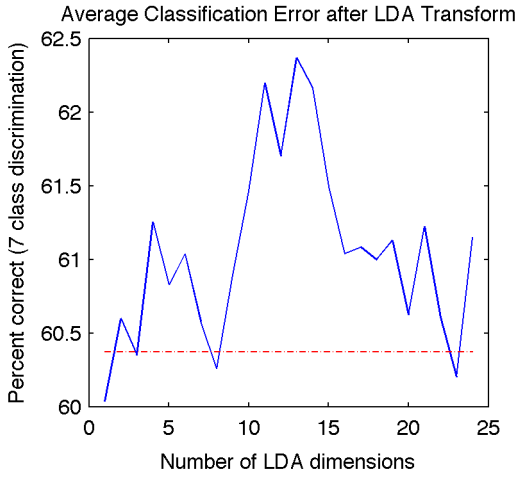

Figure 4. Genre classification performance as a function of the number of dimensions. The dashed line shows the performance without LDA.

Finally, Figure 4 shows the performance of a 7-way classifier predicting the genre labels as we vary the LDA analysis between 1 and 24 dimensions. All genre classifiers are operating well above chance; with a broad peak around 11 features. Thus we chose 11 LDA dimensions for the rest of our work.

To characterize playlist diversity we combine these three steps to convert a musical selection into a point in genre-space. The feature transformations are: 1) mean and

standard-deviation normalization, 2) SVD rotation with no dimensionality reduction, and then 3) a final rotation into an 11-dimensional space derived from a single LDA analysis using all the genre data as training data. From the points in genre space we can characterize a user's diversity.

5. Diversity

We characterize the diversity of a playlist by fitting a Gaussian-probability model to the data. A Gaussian probability surface models the data so that 63% of the data points fall within one standard deviation of the mean. We use a diagonal covariance model, estimating the variance in 11 different directions, since in most cases we do not have enough musical samples in a playlist to estimate a full 11x11 covariance matrix.

The volume of an ellipsoid is proportional the product of the length of each axis. We use the $\log_{10}$ of this volume as a measure of musical diversity. By this measure, the volume of our entire musical database, all 39k songs on the playlists, is 5.1E-12 or the $\log_{10}$ volume is -11.3. We also fit a full-covariance model to this data and the volume was smaller, indicating a better fit because the ellipsoid is not aligned with the axis and thus the feature dimensions are not fully independent.

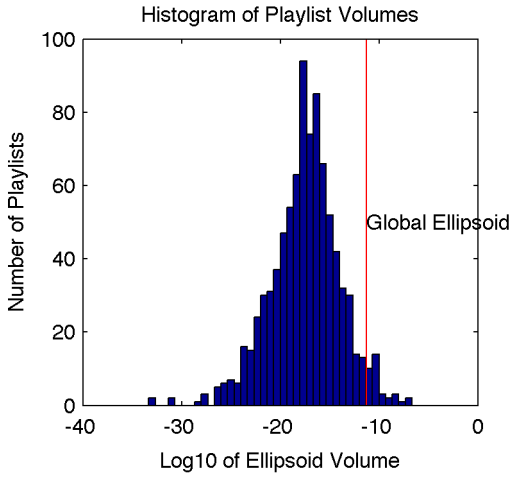

Figure 5. Histogram of playlist volume compared to the global size of the song database (red vertical line).

Figure 5 shows our basic result. A total of 887 playlists had more than 11 songs and we could reliably estimate the parameters of an 11-dimensional Gaussian. This figure compares a histogram of playlist volume to the global database maximum. There is a broad peak around a $\log_{10}$ volume of -17. On average by our measure of playlist diversity, a playlist is about 5 orders of magnitude smaller in volume than the full database.

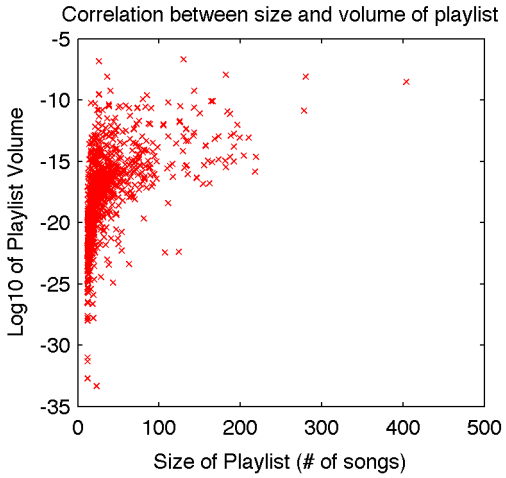

Figure 6. Individual genre volumes compared to global volume (red horizontal line).

Figure 6 shows how playlist volume compares to the size of our genre-labeled data. In general, a genre is bigger than a playlist—most all genres fall to the right of the peak in Figure 5—but are smaller than the whole database. Drums and Bass are a notable exception, perhaps because these songs are at the extremes of our GenreSpace.

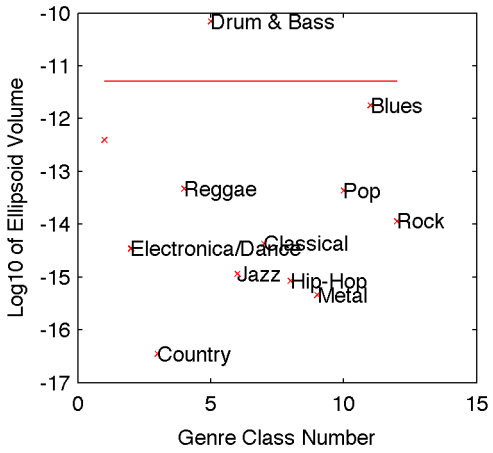

Figure 7. Correlation between size and volume of playlist.

Figure 7 shows the correlation between the number of songs in a playlist and the diversity (or volume) of a playlist. Not surprisingly, there is a significant correlation between the number of songs in a playlist and its volume. When talking about a user's taste for diversity, the length of listening time is important. This result suggests that users want more diverse suggestions as they have more time to listen to music.

6. DISCUSSION

We have presented a means to characterize the diversity of a user's musical interests. We used a large collection of manually-created playlists (887) that spanned more than 28k distinct songs. Each song was analyzed using a feature set that was designed to effectively separate different genres from each other. Each acoustic sample is represented as a single point in an 11-dimensional genre space. The distribution of points in this genre space is a measure of the playlist's diversity. We take this as evidence for users' interest in diverse music.

This argument is based on the three related hypothesis: 1) genre is an acoustically meaningful measure of music, 2) that we can use a genre-recognition task to tune the parameters of our feature space, and 3) songs that are close in genre-space sound similar to human listeners. None of these assumptions is perfect. But in the end, we only require a means to characterize whether a song falls within any given user's comfort zone. The measure can be flawed, as long as the numbers it produces are consistent within a user's expectations.

Understanding the diversity of a user's interests allows recommendation systems to generate a broader range of more relevant choices for each user. A recommendation system could pick songs based on a probability distribution defined by the variances learned from a user's playlists. The diversity varies in each dimension. This will undoubtedly work better than a system that has a single diversity limit in all dimensions.

7. Future Work

This is only the first step in a larger study to understand user's breadth of musical interests.

There are many other ways to build a vector space for music. Genre is essentially a marketing label, not a description of audio content, so some other means might be better for characterizing song similarity. One likely possibility is to calculate song similarity using user's ratings of songs—two songs that have similar ratings across the user population are probably quite similar and should be placed close together.

We also have detailed logs of what people actually listen to, and how they rate this music. We can use this data instead of playlists to characterize a user's breadth of musical interests. But this data is not public so it will be hard for researchers to compare systems. In addition, there are many other approaches to measuring the diversity of a set of points. We chose ellipsoidal volume because it has a simple basis in work on Gaussian mixture models.

But perhaps most importantly, we need to objectively compare approaches for measuring song-similarity and playlist-diversity approaches. MIREX is tackling the song-similarity problem. Measures of playlist diversity probably require asking independent raters to subjectively compare two lists of songs for diversity.

8. Acknowledgements

We appreciate the tremendous assistance we have received from George Tzanetakis and for his work on Marsyas. We also appreciate many fruitful discussions we had with Dennis Decoste, Deepak Agarwal, Ben Marlin and Lucas Gonze.

References

[1] K. Bradley and B. Smyth. Improving Recommendation Diversity. In D. O'Donoghue, editor, Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science (AICS-01), Maynooth, Ireland, pp. 75–84, 2001.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines (version 2.82), 2006. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] J. Duchene and S. Leclercq. An Optimal Transformation for Discriminant Principal Component Analysis. In IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, November 1988.

[4] J. Kleinberg and M. Sandler. Using Mixture Models for Collaborative Filtering. Proceedings of the 36th annual ACM Symposium on Theory of Computing, Chicago, IL, USA, pp. 569–578, 2004.

[5] B. Logan, D. P. W. Ellis, and A. Berenzweig. Toward Evaluation Techniques for Music Similarity. In Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR'03), Bartimore, MD, USA, 2003.

[6] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Improvements of audio-based music similarity and genre classification. In Proceedings of ISMIR, 628–633, 2005.

[7] John C. Platt, Christopher J. C. Burges, Steven Swenson, Christopher Weare, Alice Zheng. Learning a Gaussian Process Prior for Automatically Generating Music Playlists. Advances in Neural Information Processing Systems 14, pp.1425–1432, 2002.

[8] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, Vol, 10, No. 5, pp. 293–302. July 2002.

[9] George Tzanetakis and Perry Cook. MARSYAS: A Framework for Audio Analysis. In Organized Sound, Cambridge University Press, 4(3), 2000.

[10] Cheng Xiang Zhai and John Lafferty. A risk minimization framework for information retrieval. Information Processing and Management (IP&M), 42(1), Jan. 2006. pages 31–55.

[11] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan , Georg Lausen. Improving recommendation lists through topic diversification. Proceedings of WWW, 2005, pp. 22–32.