CS 74 Project Write-Up: Music Classification

Sebastian Bierman-Lytle, William Wang, James Drain

Introduction

Music can be classified in many different ways. The most common method is by genre, but many other attributes, like key signature and beats per minute, as well as both metaphorical and emotional adjectives like "uplifting" and "heavy" can be used to convey meaningful information. Software that can automatically categorize music by genre and other characterizations is becoming increasingly essential to music streaming services, which offer curated playlists to users with personalized preference settings, and to production labels and DJs, who regularly sort through thousands of new tracks to find music that exhibits some specific non-genre related qualities.

We propose to design and implement a process to automatically classify Electronic Dance Music (EDM) tracks by genre, mood, existence of vocals, gender of vocals, average pitch, and fullness, and to also assign non-mutually exclusive adjectives. One application for this process would be a music management software that automatically sorts given tracks into predefined playlists. Some softwares like this exist (such as Mixed In Key), but none currently provide emotional or adjectival classification. Products such as Pandora, Spotify, and last.fm use predictive algorithms to reccomend users songs based on their preferences, but their methods rely primarily on manual song classification and social networks inferences (what their friends like) in order to make these recommendations.

We plan to apply machine learning techniques to this problem, and extract features directly from the music waveform, so that our application can be used effectively with new and unknown data. As a final refinement, we intend to build an automated process by which a user can classify his entire library into a set of relevant playlists.

Prior Research

Tzanetakis discusses extraction vectors and formulas that are useful for preprocessing, specifically: the cetroid, flux, rolloff, zerocrossings, and lowenergy. Interestingly, these features can be drawn from 20 millisecond samples using the ShortFastFourierTransform. Other features he introduces include: full wave rectification, autocorrelation, downsampling, low pass filtering, and normalization. The algorithm itself is called automatic hierarchical genre classification.

A shortcoming to improve on is that the classifications are discrete, whereas we plan to use real-valued classifications. Also, Tzanetakis classifies solely based on genre (despite extracting other features that would be useful to us) and has a lackluster success rate (for jazz, in particular, only 37 of 98 songs were correctly categorized.)

Pachet takes his algorithm, co-ovvurrence analysis, from linguistics. Co-occurrence analysis builds a matrix of distances between songs based on what fraction of the time they are played directly after each other, using smaller weights for larger intervals between playings. Pachet then uses these distances for clustering, which vary between 50 and 80% when tested on a small sample of 12 tracks matched to 100 artists.

Interestingly, Pachet uses radio programs and cd's for data -- he uses the radio stations in order to take advantage of station themes, like classical, and to take advantage of the DJ's expertise.

Mandel and Ellis utilize support vector machines and take into account the entire song, rather than immediately mapping down to the standard set of traits. By using the entire song, their svm outperforms svm's on simplified song features by 15% (tested on an 18 genre cluster.) Mandel and Ellis also discuss differences between Kullback Leibler distance between Gaussians (which achieves the best testing rate of 69%), and Mahalonobis distance (which scores 84% when testing data included training data), and they compare those results to the success of the MFCC statistics from Tzanetakis.

Under future work, Mandel and Ellis consider using hidden Markov models to further take advantage of temporal components.

Jeremic's article is actually a tutorial for how to make a music classifying neural network. For each song, she recommends using: duration of song, tempo, root mean square (RMS) amplitude, sampling frequency, sampling rate, dynamic range, tonality and number of digital errors, and then categorizing into classical, jazz, rock, or folk. The next step is to make an account on neuroph.sourceforge.net. This is probably not a good avenue to pursue, but the resulting neural network does achieve 98% accuracy and walks through backpropagation and hidden layers, among other neural network techniques.

Chu: An interesting fact is that Chu gives is that untrained humans can correctly genre a song with 72% accuracy given only three seconds of sampling. He also very explicitly gives how to go from the waveform to MFCC features, which is based on how humans measure equal increases in pitch, and are also especially useful in quantifying pitch-independent features like formants. Chu also discusses the music-specific features first mentioned in Tzanetakis, but here in a more intuitive manner. Chu points out that Pandora utilizes unsupervized clustering to draw subtler similarities from different genres. He also reiterates that SVM's with KL are the current best classifiers with 80% accuracy. Chu acknowledges that humans like having fuzzy boundaries for genres (e.g. rock, but tetchno'y), and SVM's might have some trouble by being too discrete. Finally, Chu points towards MIREX, which has plenty of competitions in music classification, plenty of data for those contents, and also data on how competing programs have succeeded on many different genres.

Works Cited in Research

Tzanetakis, George, George Essl, and Perry Cook. (2001). "Automatic Musical Genre Classification of Audio Signals." Accessed October 2, 2014. 2nd Annual International Symposium on Music Information Retrieval 2001.
http://ismir2001.ismir.net/pdf/tzanetakis.pdf

Pachet, F. Westermann, G. Lairgre, D. (2001). Musical Data Mining for Electronic Music Distribution, WedelMusc, Firenze (It). Accessed October 2, 2014. Sony Computer Science Laboratory Paris 2002-2014.
http://www.csl.sony.fr/downloads/papers/2001/pachet01c.pdf

Mandel, Mitchell I., and Daniel P.W Ellis. (2005). "Song-Level Features and Support Vector Machines for Music Classification." Accessed October 2, 2014. 2014 Columbia University.
http://www.ee.columbia.edu/~dpwe/pubs/ismir05-svm.pdf

Jeremic, Marina. "Music Classification by Genre Using Neural Networks." Accessed October 2, 2014. Sourceforge:
http://neuroph.sourceforge.net/tutorials/MusicClassification/music_classification_by_genre_using_neural_networks.html

Chu, Mei-Lan. "Automatic Music Genre Classification." Accessed October 2, 2014. Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan
http://djj.ee.ntu.edu.tw/music_genreclass.pdf

Model

Preliminarily, we intend to apply a multivariate gaussian mixture model to classify ("tag") each song. This model is intended to receive a song (in the form of raw waveform data) and generate a list of tags which apply to the song.

First, we will process the training set in a supervised learning process. Each track processed in this fashion will have a hand-picked list of tags. Any subjectivity in this process is intentional, as in the ideal final product the user can customize the algorithm to suit their own personal tastes by supplying a new training set.

The program reads and computes the features for each song. By iterating through a sufficiently large training set, it can estimate a gaussian distribution for each feature, conditioned on each tag. For example, it may find that "happy" songs tend to exhibit a considerably higher tempo (as measured in bpm) than "sad" songs, so the distributions conditioned on each tag for this feature will be noticeably different. Note that a gaussian distribution may not be the ideal representation of every feature. However, due to ease of use / calculation it will serve at least as a starting point. See below ("Data") for more detail on the computation of these features.

Secondly, the user will supply new tracks to be classified automatically. Having developed a clear characterization of each tag in the learning process, the actual calculations (assuming each feature can ultimately be condensed to a single scalar variable) are quite straightforward. As expressed in pseudocode:

```
$D$ ← Matrix of probability distributions from training set
for each song $s$:
    $T$ ← Vector of tags applied to song
    for each tag $t$:
        $F$ ← Vector of song's features
        $P$ ← Probability($F$,$D$)
        If $t \in$ adjectives:
            If WeightedAvg($P$) > $THRESHOLD$:
                $T(t)$ ← True
        Else:
            If $t$ = max(group(t)):
                $T(t)$ ← True
                $T(t\_OldMax)$ ← False
```

Essentially, the joint probability of the song's feature vector is calculated, conditioned on each tag's data set. If the tag is an adjective (see below), the tags are not mutually exclusive and any tags for which that probability exceeds some threshold will be tagged accordingly. If the tag is not an adjective, its group is mutually exclusive and the highest probability tag is selected. The computation of probability can be summarized as follows:

$$P(F|t) = \sum_{i=1}^{K} P(F_i, \mathcal{N}(\mu_{i,t}, \sigma_{i,t}))$$

Ultimately, a playlist will be created for each tag in the set, containing all the songs matching that tag. Unfortunately, the current standard for classification (ID3 tags) does not support an arbitrary list of tags on the file itself. Nevertheless, the resulting list of playlists should enable the program to directly contribute to any music selection process.

Data

Overview
A custom dataset will be created for the project, so that the samples can be curated to reflect the range of music genres we are interested in analyzing and contain the specific outputs that we are searching for. Samples will be taken from online sources and the group members music libraries (we have about 10,000 EDM tracks to work with). To start, 3 tracks from each distinct categorization will be used in the data set. After the first milestone, we will increase the data set and test boundary cases (such as testing on a dubstep track that is at the BPM of a moombah track).

There are some publicly available libraries of large sets of audio classification data, but they are all too general in nature – for instance, all electronic music is usually categorized as "electronic" and we would like to distinguish at least a dozen different sub genres within that category.

Creating this data set will be a two-step process: first, we will write a Matlab script that accepts the amplitude data of a music track and outputs a vector containing features that define that track; second, we will pair a vector of handcrafted tags with each feature vector. These pairs of vectors will then become the inputs and outputs of our machine learning algorithm.

<u>Data Feature Specifics</u>
The data extracted from each music track will be comprised of various audio features. The final set of features will be determined by whichever combination leads to the best categorization, but we will definitely be utilizing beats per minute, key, and data derived from MFCCs. We will also derive features from the distributions of frequency across time segments, with an emphasis on the chorus/drop segment, such as average frequency content, rate of change between frequency ranges, and amplitude of low frequencies in offbeat segments.

This kind of analysis is essential for differentiating sub genres, since the majority of distinctions can only be found in the sounds played between beats. In many scenarios, data derived from averaging values across time is only useful if the track is segmented into distinct musical sections. Part of the challenge will be to determine when data in the music track is important to its classification and when it is not.

Since the result portion of our data is subjective, the number and nature of these derived features will be determined through research and experimentation as new challenges in classification and differentiation arise.

<u>Data Output Specifics</u>
The tags that we define as results will be grouped into categories and evaluated separately. The groups of categories are as follows: genre, mood, existence of vocals, gender of vocals, average pitch, fullness, and adjectives. The algorithm will assign one value to each music track for each of these categories, except for adjectives, which is a unique group for non-exclusive traits. A list and description of the possible outcomes follows:

*Genre***:** Deep House, Tech House, Minimal House, Melbourne House, Dutch House, Big Room House, Progressive House, Tropical House, Folk House, Swing House, Dance, Electro, Hardstyle, Hard Techno, Trance, Moombah, Dubstep, Glitch Hop, Trap, Twerk, Downtempo, RNB, Rap

*Mood***:** Uplifting, Nostalgic, Happy, Sad, Angry, Neutral

- The emotional quality of the music

*Existence of Vocals***:** Yes, No

Gender of Vocals: Male, Female

Average Pitch: Bass Heavy, Mid Range, High Pitch

- Describes if the music is mostly comprised of lower, higher, or mid range frequencies.

Fullness: Very Minimal, Minimal, Mid Range, Saturated

- Describes the fullness of the frequency spectrum (ei. the number of distinct instruments/sounds used at once).

Adjectives: Anthem, Big, Chill, Bouncy, Dark, Light, Deep, Dreamy, Grimey, Heavy

- Subjective adjectives given to music that are useful in describing tangible qualities in a metaphorical context.

Project Milestone

For our project milestone, we aim to complete the following:

- Complete a Matlab script that reads an audio file and computes all the features that we wish to experiment with

- Finalize the model for our supervised learning process

- Create a dataset with at least 3 tracks exhibiting each genre/quality (minimum of 70 tracks)

- Implement the supervised learning process so that we can begin evaluating the features