Automatic Music Clustering using Audio Attributes

Abhishek Sen
BTech (Electronics)
Veermata Jijabai Technological Institute (VJTI),
Mumbai, India
abhishekpsen@gmail.com

Abstract—*Music brings people together, it allows us to experience the same emotions. Currently musical genre classification is done manually and requires even the trained human ear considerable effort. Therefore, clustering songs automatically and then drawing valuable insights from those clusters is an interesting problem and can add great value to music information retrieval systems. Most of the work in this field has involved extracting the audio content from audio files. This paper explores a novel technique to cluster songs based on Echonest Audio Attributes and K-Means algorithm. I experiment with different sets of attributes and genres. Most notably, I achieve 75-85% accuracy on a 4 genre-dataset of Classical, Rap, Metal and Acoustic songs. This result is better than results reported for human song clustering.*

Keywords- Music clustering; K-Means; Musical genre classification; Song recommendation; Echonest

I. Introduction

Music is a language that everyone understands and enjoys, it is universal. A music genre is a category that identifies pieces of music as belonging to a shared tradition or set of conventions like certain styles or a 'basic musical pattern'. Most often, music can be categorized based on what feeling it evokes. For instance, rock music evokes more energy with its fast beat patterns and rhythm, while classical music is softer, more profound, soothing to listen to. Music can also be categorized at times based on underlying themes or geographic origins.

Clustering songs into genres is useful because, we develop strong tastes in particular genres, as per our liking. It helps us organize our music collection and in electronic music distribution. A large amount of work is done nowadays to improve music recommendation systems. Music streaming websites like Grooveshark or Pandora, online retailers like Amazon, give us suggestions based on what kind of music we listen to or buy. Current recommendation systems mainly use manual genre-annotations which is time-consuming and expensive, or collaborative filtering, but there are several underlying flaws with this technique. (Eg: when spammers give wrong reviews or when we recommend a song for someone else and the recommendation system takes that into account as our own taste, etc.) Music recommendation systems can be improved a lot by algorithmic music clustering.

II. Related Past Work

Most research in the past has involved extracting musical features from audio files, or by studying the raw waveform and applying signal processing techniques to find a close match with specific genres. Tzanetakis et al. [1] used timbral texture, rhythmic content and pitch content as features to train statistical pattern recognition classfiers with 61% accuracy.

Danny Diekroeger [2] tried to classify songs based on the lyrics using a Naïve Bayes classifier, but concluded that analyzing solely lyrics is not enough.

Panagakis et al. [3] used multiscale spectro-temporal modulation features, inspired by an auditory cortical processing model.

However, these techniques haven't been very accurate as they fail to incorporate subjective features like liveliness, energy and so on. Grouping music is inherently a subjective task and it is important to translate those subjective features into some quantifiable measure.

III. Feature extraction and datasets

As mentioned, music genre classification research so far has mainly focused on spectro-temporal features. A way of measuring subjective features was missing. Echonest [4] is an online music information repository, containing information about 30 million songs from 1.5 million artists. It has a neat developer API written in python (PyEchonest) that lets one pull an audio_summary object for each song. This object has a measure of subjective features, some of which are as follows:

TABLE I.

Feature	Description	Range
Time_Signature	An estimated overall time signature	Integer
Energy	The song makes you feel energetic or dull	0-1
Danceability	Ease with which one can dance to a song, including beat strength, tempo	0-1
Tempo	the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration	0-1
Duration	the duration of a track in seconds as precisely computed by the audio decoder	Float
Key	the estimated overall key of a track. The key identifies the tonic triad, the chord, major or minor, which represents the final point of rest of a piece	Integer

a. Echonest Features.

These features have been calculated with data from track analysis and from inputs of EchoNest's Data QA Team, many of whom are qualified musicians [5].

The data was gathered using a python script, downloading the above features and five more features (liveness, speechiness, instrumentalness, loudness, acousticness) for seven genres: Classical, Rap, Metal, Acoustic, Rock, Reggae and Indian Classical. Around 500 songs of each genre were chosen randomly and their respective features were gathered using the PyEchoNest API. All the data was written to .dat files, that can be easily extracted and processed in the data processing step.

IV. EXPERIMENTAL SETUP

K-Means algorithm was used to cluster similar songs together. Euclidean distance was used as the distance measure. Experiments were performed with varying sets of genres and with varying number of datapoints (100,150,.. 400 songs of each genre) to analyze the performance of the algorithm. K-Means is an iterative process that proceeds in two steps namely cluster assignment and compute centroids.

In the cluster assignment" phase of the K-means algorithm, the algorithm assigns every training example x(i) to its closest centroid, given the current positions of centroids. Specifically, for every example i we set

$c^{(i)} :=$ j that minimizes $\| x^{(i)} - \mu_j \|^2$

where $c^{(i)}$ is the index of the centroid that is closest to $x^{(i)}$, and j is the position (value) of the j'th centroid**.**

Given assignments of every point to a centroid, the second phase of the algorithm recomputes, for each centroid, the mean of the points that were assigned to it. Specifically, for every centroid k we set

$\mu_k := (\sum x^{(i)}) / (| C_k |)$

where C_k is the set of examples that are assigned to centroid k.

As K-Means faces the problem of local optimum, several iterations was run with randomly initialized seed centroids, calculating the cost function (error) each time, and taking the centroid set with minimum error as the final answer. Also, all features were normalized before running K-Means.

The entire algorithm was implemented in GNU Octave.

V. RESULTS

The program was first run on a dataset of two genres.

Accuracy was defined as usual $\frac{\text{No of songs labelled correctly}}{\text{Total no of songs}}$

In the first run, a dataset of 400 songs was taken with 200 Classical songs in the first 200 lines and 200 Rap songs in the next 200 lines. Histograms were plotted for two bins, Cluster label frequencies for the first 200 and for the next 200. In the ideal case, both the histograms should have unique peaks of 200 each. As can be seen, the algorithm labels 195 songs as cluster 2 in the first 200 songs, and 199 songs as cluster 1 in the next 200 songs.

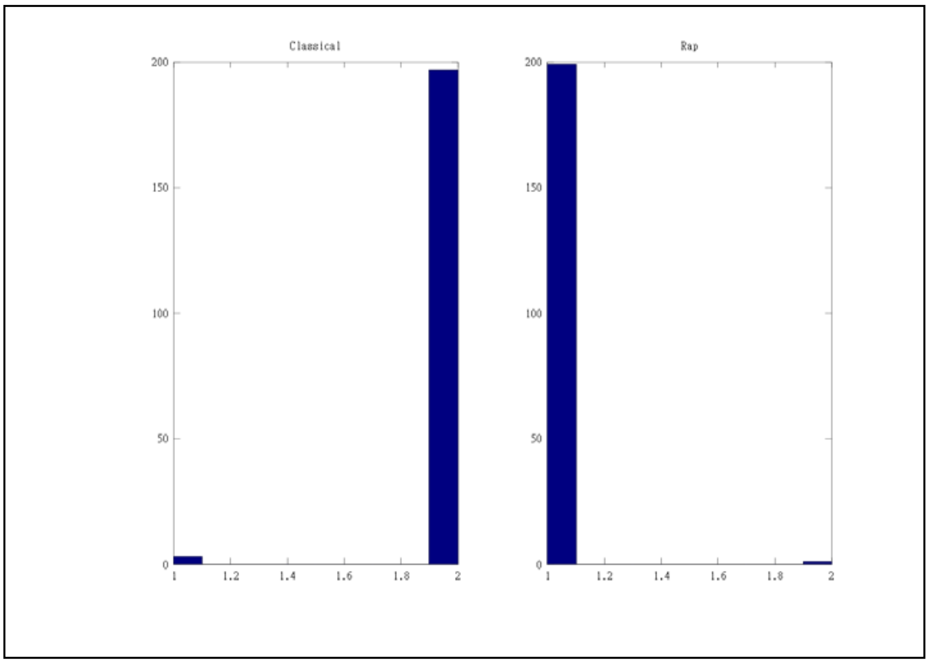

Figure 1. Histogram Classical vs Rap

$$\text{Accuracy} = \frac{394}{400} = 98.5\%$$

In the next step, a third genre: Metal was added to the dataset. 200 songs of Classical, 200 songs of Rap, 200 songs of Metal.

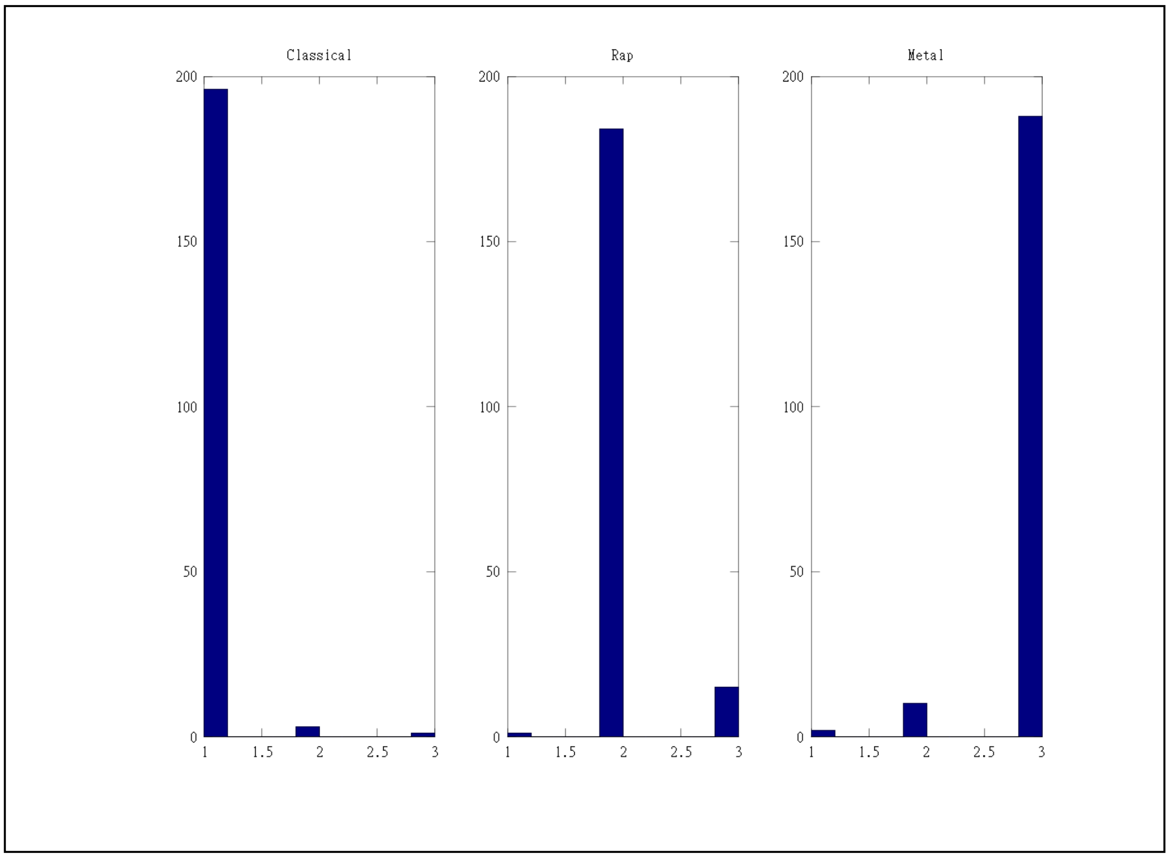

Figure 2. Histogram Classical vs Rap vs Metal

The first histogram is for the first 200 songs (that are Classical), 196 songs are put labelled as cluster 1.Iin the next 200 songs (that are Rap), 183 songs are labelled as cluster 2 and in the last 200 (that are Metal), 192 songs are labelled as cluster 3. Thus we have three clean unique peaks, the algorithm performs decently.

$$\text{Accuracy} = \frac{571}{600} = 95.16\%$$

Next, clustering is tried on 4 genres: Classical, Rap, Metal, Indian Classical

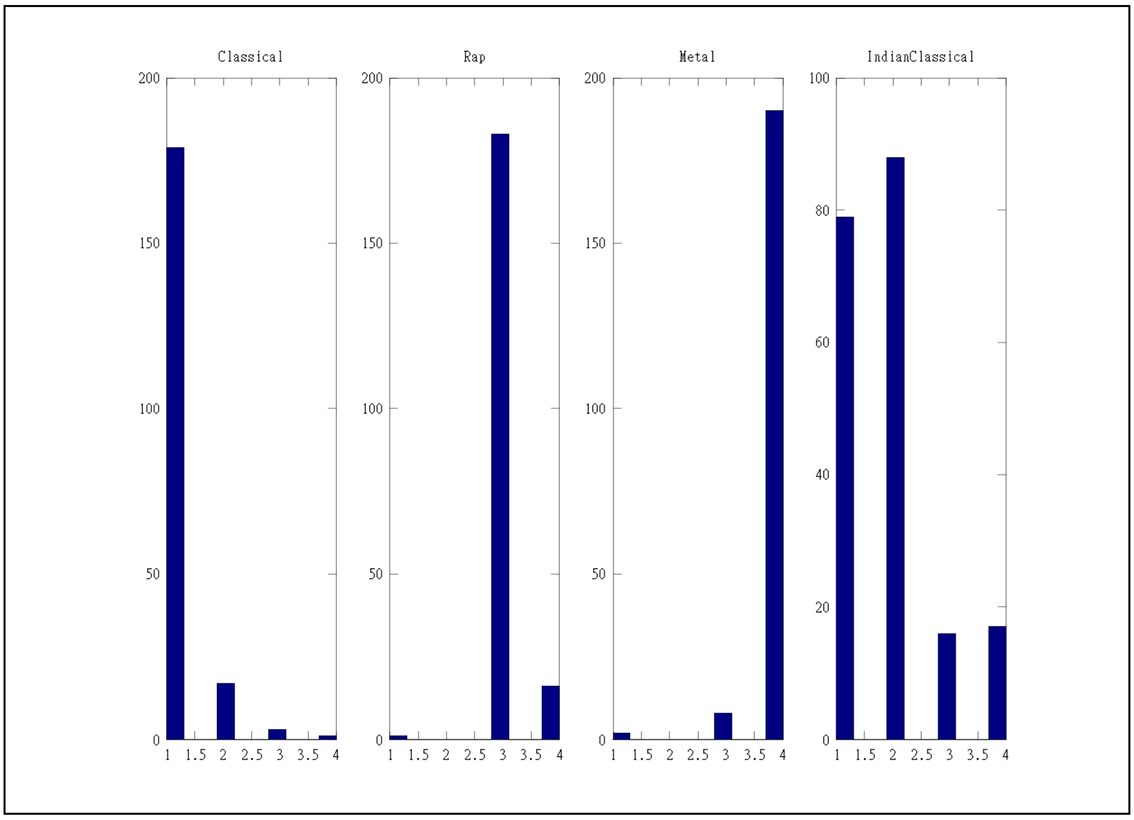

Figure 3. Histogram Classical vs Rap vs Metal vs Indian Classical

I don’t get very clean clustering in this case. The first 200 songs (Western Classical) has a peak of label 1 and the last 200 songs (Indian Classical) has peaks of 2 and 1, indicating that a lot many Indian Classical songs have been clustered along with Classical songs. This observation is in accordance with the musical insight that there are many parallels between Western Classical and Indian Classical music [6].

Next, we take datasets of varying sizes (100-400) of 4 genres: Classical, Rap, Metal, Acoustic and plot Accuracy vs No. of datapoints, to gain an insight about how the size of the dataset affects accuracy.

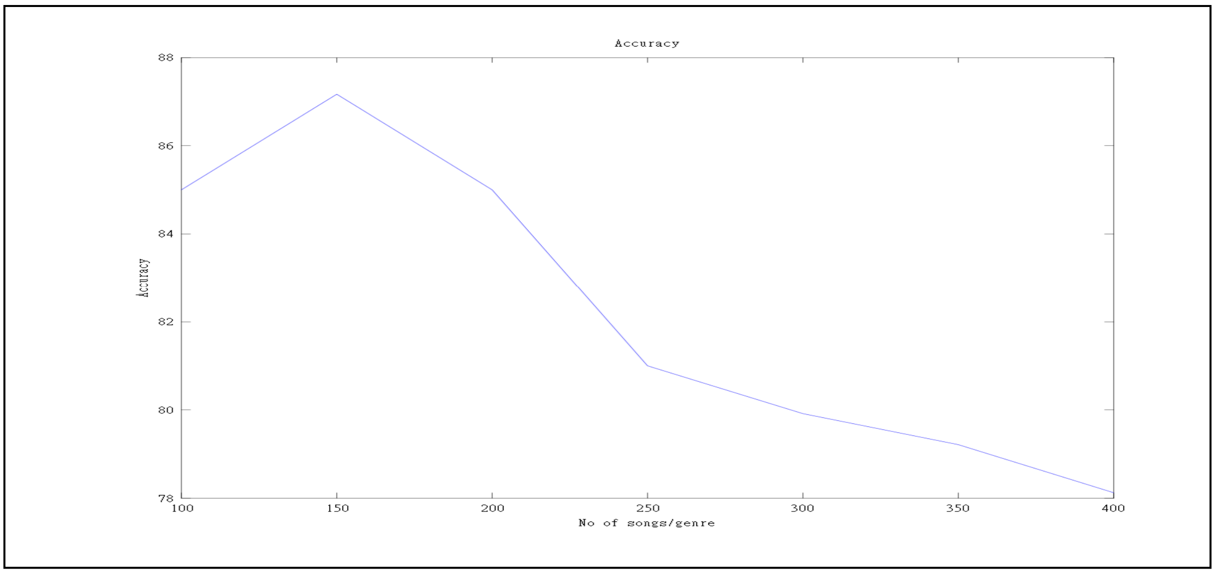

Figure 4. Accuracy vs No of songs per genre

When run on datasets of sizes varying from 400 to 1600 songs (having equal no of songs of each genre), the accuracy ranges from 87% to 78%. We can see, the algorithm gradually becomes less accurate as the size of dataset increases, but still gives a considerably good accuracy.

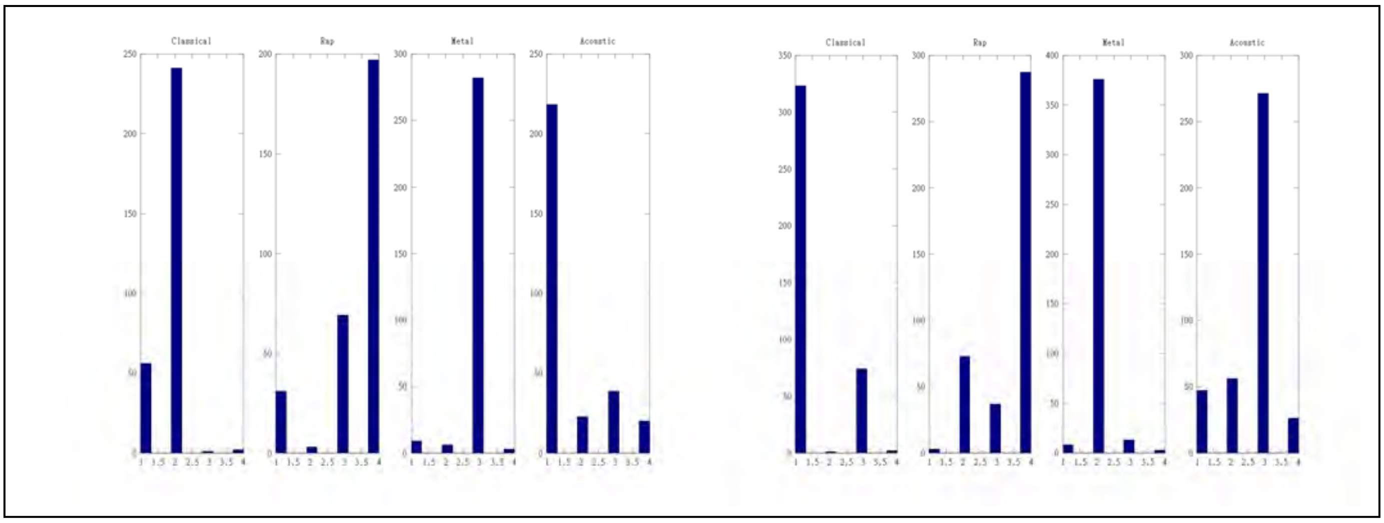

Figure 5. Histograms for datasets sise 1200 and 1600 (Classical vs Rap vs Metal vs Acoustic)

I get clean and unique peaks for the same 4 genres (Classical vs Rap vs Metal vs Acoustic).

All code and datasets used are available on my GitHub account (https://github.com/abhishekpsen/KMeans-PyEchoNest).

VI. FUTURE WORK

K-Means with EchoNest Audio features clearly beats manual tagging (reported to be around 70% [7]).

This indicates that the underlying idea behind this is sound. An interesting extension to this method is to use more features, adding timbral texture information, and using Natural Language Processing techniques to detect the underlying theme from the lyrics. It is to be seen how well the technique scales for very large datasets.

This technique is suitable for music recommendation frameworks, by preclustering songs, pulling in personal musical tastes using Facebook API, detecting the cluster, and then recommending songs from the same cluster.

VII. REFERENCES

[1] Musical Genre Classification of Audio Signals, George Tzanetakis and Perry Cook, IEEE Transactions on Speech and Audio Processing,

[2] Can Song Lyrics Predict Genre?, Danny Diekreoger, http://cs229.stanford.edu/proj2012/Diekroeger-CanSongLyricsPredictGenre.pdf

[3] Music genre classification: A multilinear approach, Y. Panagakis, E. Benetos, C. Kotropoulos, ISMIR 2008 – Session 5a

[4] Echonest http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf

[5] Echonest feature defintions, http://runningwithdata.com/post/1321504427/danceability-and-energy

[6] Similarities between Western and Indian Classical music, http://www.itcsra.org/sra_faq_index.html

[7] 'Scanning the Dial: The Rapid Recognition of Music Genres', Robert o. Gjerdingen and David Perrot Journal of New Music Research,

[8] Vol. 37, No. 2. (2008), pp. 93-100