Music Mood Classification Using The Million Song Dataset

Bhavika Tekwani

December 12, 2016

Abstract

In this paper, music mood classification is tackled from an audio signal analysis perspective. There's an increasing volume of digital content available every day. To make this content discoverable and accessible, there's a need for better techniques that automatically analyze this content. Here, we present a summary of techniques that can be used to classify music as *happy* or *sad* through audio content analysis. The paper shows that low level audio features like MFCC can indeed be used for mood classification with a fair degree of success. We also compare the effects of using certain descriptive features like acousticness, speechiness, danceability and instrumentalness for this type of binary mood classification as against combining them with timbral and pitch features. We find that the models we use for classification rate danceability, energy, speechiness and the number of beats as important features as compared to others during the classification task. This correlates to the way most humans interpret music as happy or sad.

1 Introduction

Music Mood Classification is a task within music information retrieval (MIR) that is frequently addressed by performing sentiment analysis on song lyrics. The approach in this paper aims to explore to what degree audio features extracted from audio analysis tools like *librosa*, *pyAudioAnalysis* and others aid a binary classification task. This task has an appreciable level of complexity because of the inherent subjectivity in the way people interpret music. We believe that despite this subjectivity, there are patterns to be found in a song that could help place it on Russell's [1] 2D representation of valence and arousal. Audio features might be able to overcome some of the limitations of lyrics analysis when the music we aim to classify is instrumental or when the song spans many different genres. Mood classification has applications ranging from rich metadata extraction to recommender systems. A mood component added to metadata would make for better indexing and search techniques leading to better discoverability of music for use in films and television shows. Music applications that enable algorithmic playlist generation based on mood would make for richer, user-centric applications. In the next few chapters, we discuss the approach that leads us to 75% accuracy and how it compares to other work done in this area.

2 Problem Statement

We aim to achieve the best possible accuracy in classifying our subset of songs as *happy* or *sad*. For the sake of simplicity, we limit ourselves to these two labels though they do not sufficiently represent the complex emotional nature of music.

2.1 Notations

We introduce some notations for the feature representations in this paper.

$$f_{timbre_avg} = [timavg_1, timavg_2...timavg_{12}] \tag{1}$$

(1) represents the vector of timbral average features at the song level.

$$f_{pitch} = [pitch_1, pitch_2...pitch_{12}] \tag{2}$$

(2) represents a vector of chroma average features at the song level.

$$f_{timbre} = [tim_1, tim_2...tim_{90}] \tag{3}$$

(3) is a vector of mean and covariance values of all the segments aggregated at the song level.

3 Literature Review

3.1 Automatic Mood Detection and Tracking of Music Audio Signals (Lie Lu et al)

Lie Lu et al [3] explore a hierarchical framework for classifying music into four mood clusters. Working with a dataset of 250 pieces of classical music, they extract timbral Mel Frequency Cepstral Coefficients (MFCC) and define spectral features like shape and contrast. These are used in the form of a 25 dimensional timbre feature. Rhythm features are extracted at the song level by finding the onset curve of each subband (an octave based section of a 32 ms frame) and summing them. Calculating average correlation peak, ratio between average peak strength and average valley strength, average tempo and average onset frequency leads to a five element rhythm feature vector. They use the mean and standard deviation of the frame level features (timbre and intensity) to capture the overall structure of the frame.

A Gaussian Mixture Model (GMM) with 16 mixtures is used to model each feature related to a particular mood cluster. The Expectation Maximization (EM) algorithm is used to estimate the parameters of Gaussian components and mixture weights. In this case, K-Means is used for initialization. Once the GMM models are obtained, the mood classification depends on a simple hypothesis test with the intensity features given by the equation below.

$$\lambda = \frac{P(G_1/I)}{P(G_2/I)}, \begin{cases} \geq 1, Select \quad G_1 \\ < 1, Select \quad G_2 \end{cases} (4)$$

Here, λ represents the likelihood ratio, $\mathrm{G_i}$ represents different mood groups, I is the intensity feature set and $\mathrm{P(G_i| I)}$ is the probability that a particular audio clip belongs to a mood group $\mathrm{G_i}$ given its Intensity features which are calculated from the GMM.

3.2 Aggregate Features and ADABOOST for Music Classification (Bergstra et al)

Bergstra et al [10] present a solution towards artist and genre recognition. Their technique employs frame compression to convert frames from songs to a song level set of features based on covariance. They borrow from West & Cox [8] who introduce a "memory feature" containing the mean and variance of a frame. After computing frames, they group non-overlapping blocks of frames into segments. Segment summarization is done by fitting independent Gaussian models to the features. Covariance between the features is ignored. The resulting mean and variance values are inputs to ADABOOST. Bergstra et all explore the effects of varying segment lengths on classification accuracy and conclude that in smaller segments, mean and variance of the segments have higher variance.

3.3 An Exploration of Mood Classification in the Million Songs Dataset (Corona et al)

Corona et al [11] perform mood classification on the Million Song Dataset using lyrics as features. They experiment with term weighting schemes like TF, TF-IDF, Delta TF-IDF and BM25 to explore the term distributions across four mood quadrants defined by Russell[1]. The Kruskal Wallis test is used to measure statistically significant differences in the results obtained using different term weighting schemes. They find that a support vector machine (SVM) provides the best accuracy and moods like *angst*, *rage*, *cool-down* and *depressive* were predicted with higher accuracy than others.

3.4 Music Mood Classification

Goel & Padial [9] attempt binary classification for mood on the Million Song Dataset. They use features like Tempo, Energy, Mode, Key and Harmony. The harmony feature is engineered as a 7 element vector. A soft margin SVM with the RBF kernel is used for classification to provide a success rate of 75.76%.

3.5 Music Genre Classification with the Million Song Dataset

Liang et al [5] use a blend model for music genre classification with feature classes comprising of Hidden Markov Model (HMM) genre probabilities extracted from timbre features, loudness and tempo, lyrics bag-of-words submodel probabilities and emotional valence. They assume each genre corresponds to one HMM and use labeled training data to train one HMM for each genre. Additionally, they combine audio and textual (lyrics) features for Canonical Correlation Analysis (CCA) by revealing shared linear correlations between audio and lyrics features in order to design a low dimensional, shared feature representation.

4 Methods and Techniques

4.1 Feature Engineering and Selection

For mood classification, one of the questions we try to answer is, can a model capture the attributes that make a song *happy*or *sad* the same way we as humans do? To answer this question, we used Recursive Feature Elimination (RFECV) with a Random Forest Classifier and 5-fold cross validation. Recursive Feature Elimination is a Backwards Selection technique that helps you find the optimal number of features that minimize the training error. Additionally, once we select the features we also examine the relative importance of these features for different estimators to better understand whether some features are better indicators of mood than others. We mutliplied mode and key and tempo and mode to capture the multiplicative relations between these features. Loudness is provided in decibels and is often negative, so we squared the value for better interpretability. Values for Speechiness, Danceability, Energy, Acousticness and Instrumentalness were often missing when we tried using the Spotify API to fetch them. In that case, we imputed the mean of these values.

The dataset includes two features Segments Pitches and Segments Timbre which are both 2D arrays of varying shapes. A segment is a 0.3 second long frame in a song. This means that the number of segments varies with the song. Segments Timbre is a 12 dimensional MFCC-like feature for every segment. MFCC is a representation of the short-term power spectrum of a sound obtained by taking a cosine transform of the power spectrum and converting it to the Mel scale. These are very commonly used in audio analysis for speech recognition tasks. In our dataset, the Echo Nest API's Analyze documentation [14] states that they provide Segments Timbre functions by extracting MFCC for each segment in a song and then using Principal Component Analysis (PCA) to compactly represent them as a 12 element vector. In a similar vein, Segments Pitches represent the chroma features of each segment in a 12 dimensional vector. Here, the 12 elements of the vector represent pitch classes like C, C#, B and so on. The challenge is - to find a uniform representation of timbre and pitches that represents a whole song. We use a technique called segment aggregation[8, 10, 3].

Segment aggregation involves computing several statistical moments like mean, minimum, maximum, standard deviation, kurtosis, variances and covariances across each segment. We try two methods. First, we compute a vector containing the mean and covariances of all segments and obtain a 90 element vector (12 averages and 78 covariances). We can use this approach for timbre and pitch arrays both. The drawback is that 90 elements make for a very large feature vector and they would need to be pruned in some way or the most important elements would have to be identified. Using PCA is not desirable here for two reasons: timbre features have already been extracted through PCA on MFCC values and our segment aggregation does not account for temporal relations between the segments. This leads to some loss of information in the segments. Using the 90 element vectors as they are introduces the curse of dimensionality. Our second approach is calculating only the elementwise mean of all segments in a song. This gives us two 12 dimensional vectors for pitches and timbre. Now, we use these as features for our models.

Using RFECV, we selected 12 timbre features (equation (1)), 12 pitch averages (equation (2)) and descriptive features like Danceability, Speechiness, Beats, LoudnessSq, Instrumentalness, Energy and Acousticness for a total of 31 features. Other features like Key*Mode, Tempo*Mode, Time Signature, Key and Mode were found to not aid the classification task and were discarded.

4.2 Classification Models

For this binary classification problem, we evaluate several models and compare how the perform on the test set. To tune the performance of each model, we perform a hyperparameter search and then select the ones that perform best with each model. 5 fold cross validation is used during the hyperparameter search.

Table 1 below shows the different estimators we used and the parameters we tuned for each.

Estimators	**Hyperparameters**
Random Forest Classifier	estimators= 300, max. depth = 15
XGBoost Classifier	max. depth = 5, max. delta step = 0.1
Gradient Boosting Classifier	loss = exponential, max. depth = 6, criteria = mse, estimators = 200
ADABOOST Classifier	learning rate = 0.1, no. of estimators = 300
Extra Trees Classifier	max. depth = 15, estimators = 100
SVM	C = 2, kernel = linear, gamma = 0.1
Gaussian Naive Bayes	priors = None
K Nearest Neighbour Classifier	number of neighbours = 29, P = 2, metric = euclidean

Table 1: Tuned hyperparameters for various estimators

5 Discussion and Results

5.1 Datasets

We are using the Million Song Dataset (MSD) created by LabROSA at Columbia University in association with Echo Nest. The dataset contains audio features and metadata for a million popular tracks. For the purpose of this project, we use the subset of 10,000 songs made available by LabROSA. The compressed file containing this subset is 1.8 GB in size. Using this dataset in its original form was a challenging task. We hand labeled 7396 songs as *happy* and *sad*. This was time consuming and the only hurdle to attempting hierarchical classification. We use a naive definition of *happy* and *sad* labels. Songs that would be interpreted as angry, depressing, melancholic, wistful, brooding, tense/anxious have all been tagged as *sad*. On the other hand songs interpreted as joyful, rousing, confident, fun, cheerful, humourous, silly have been tagged as *happy*. Admittedly, this is an oversimplification of the ways music can be analyzed and understood. An obvious caveat of this method is that it does not account for subjectivity in the labels and only one frame of reference is used as ground truth. However, to deal with this to some extent, we dropped songs that we couldn't neatly bucket into either labels. This means that a song as complex as Queen's Bohemian Rhapsody does not appear in the dataset. In Table 1, we present a snapshot of the data available to us and Table 2 shows the different categories our attributes fall into.

Million Song Dataset	**Spotify API**
Artist Name	Danceability
Title	Speechiness
Tempo	Instrumentalness
Loudness	Energy
Segment Pitches	Acousticness
Segment Timbre	
Beats confidence	
Loudness (dB)	
Duration (seconds)	
Mode	
Key	
Time Signature	

Table 2: Fields in the Million Song Dataset and Spotify API

Notational	**Descriptive**	**Audio**
Key, Mode,	Speechiness, Danceability, Instrumentalness,	Segment Pitches, Segment Timbre
Time Signature	Energy, Acousticness	Tempo, Beats confidence

Table 3: Attribute Categories

On downloading the dataset and inspecting it, we found that the values of Energy and Danceability which were supposed to be a part of the dataset were 0 in all the tracks. According to the Analyze documentation [14], it means these values were not analyzed. However, Energy and Danceability were crucial features that we needed for our task. To solve this problem, we used the Spotify API (the Echo Nest API is now a part of Spotify's Web API). We fetched descriptive features like Energy, Acousticness, Danceability, Instrumentalness and Speechiness for the 7396 songs.

5.2 Evaluation Metrics

The dataset contains a near equal distribution of *happy* and *sad* songs as shown in Table 4.

Label	**Train**	**Test**
Happy	2171	1522
Sad	2205	1498

Table 4: Train and test set distributions

Hence, we decide that Accuracy would be the correct metric to use.

Accuracy is defined in (5) where TP, TN, FP, FN stand for True Positive, True Negative, False Positive and False Negative.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

5.3 Experimental Results

We aim to evaluate three types of feature subsets.

In Table 5, P represents Pitch, T represents Timbre and D represents Descriptive features.

Timbre and pitch features are shown in equations (1) and (2) respectively.

Descriptive features include Danceability, Energy, Speechiness, Acousticness, Instrumentalness, Beats and LoudnessSq.

Estimator	**Features**	**Test Accuracy**
Random Forest Classifier	P, T, D	0.7456
	P , T	0.7291
	D	0.7182
ADABOOST Classifier	P. T. D	0.7354
	P, T	0.7168
	D	0.7119
XGBoost Classifier	P, T, D	**0.7533**
	P, T	0.7344
	D	0.7165
Gradient Boosting Classifier	P, T, D	**0.7552**
	P, T	0.7145
	D	0.7105
SVM	P, T, D	0.7350
	P, T	0.7142
	D	0.6966
K Nearest Neighbor Classifier	P, T, D	0.6397
	P, T	0.6725
	D	0.5360
Extra Trees Classifier	P, T, D	0.7447
	P, T	0.7245
	D	0.7178
Gaussian Naive Bayes Classifier	P, T, D	0.6821
	P, T	0.6645
	D	0.6417
Voting Classifier	P, T, D	**0.7506**
	P, T	0.7238
	D	0.7132

Table 5: Classification Accuracy by Estimator and Features

6 Conclusion

We observe from our experimental results that Ensemble classifiers like Random Forests, XGBoost, Gradient Boosting Classifier, ADABOOST perform better on our test set than SVMs and Naive Bayes classifier. Comparing our results to the work of Goel & Padial [9] we see that our highest accuracy is 75.52 % with a Gradient Boosting Classifier whereas they achieved 75.76% with an SVM using an RBF kernel. The difference in dataset size is significant as we compare our 7396 to their 233. We feel that this is a fair result but the feature extraction process can be improved. To answer the questions we ask in the problem statement, yes, audio features do aid in the mood classification task. Table 5 shows that using audio features like pitch and timbre along with descriptive features provides atleast a 3% increase in accuracy. Additionally, pitch and timbre averages themselves are sufficient to reach 72.91% accuracy with Random Forest Classifiers.

6.1 Directions for Future Work

In this music mood classification task, the lack of ground truth labels for a dataset as large as MSD was a significant hurdle to any further exploration of genre-mood relationships, canonical correlation analysis between music and lyrics or hierarchical mood classification. We attempted some analysis to understand the relation between genre and mood but we only had genre labels for approximately 2000 songs out of the 7396 we labeled. Now that we are able to achieve upto 75% test accuracy, hierarchical mood classification would be the next step if we had ground truth labels for moods that fall under *happy* and *sad*. We can demonstrate this by building a recommender system that allows you to enter a song title and then suggests a song similar to the one you entered. Similarity of songs would be based on features like emotional valence, timbre, pitch and others.

A simple framework for this would have the following steps:

1) Enter a song title based on which you want recommendations.

2) Analyse the song to assign it to a mood based cluster.

3) Suggest a song from the cluster that is most similar to the one entered by the user based on how close they are in terms of pitch, timbre, energy and valence.

References

[1] J. A. Russell, *A Circumplex Model of Effect*, Journal of Personality and Social Psychology, (6), 1980.

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. *The Million Song Dataset.* In Proceedings of the 12th Interna-

tional Society for Music Information Retrieval Conference (ISMIR 2011), 2011

[3] Lie Lu, D. Liu, and Hong-Jiang Zhang. *Automatic Mood Detection and Tracking of Music Audio Signals*, IEEE Transactions on Audio, Speech and Language Processing 14, no. 1 (January 2006): 5–18. doi:10.1109/TSA.2005.860344.

[4] Panagakis, Ioannis, Emmanouil Benetos, and Constantine Kotropoulos. *Music Genre Classification: A Multilinear Approach.* In ISMIR, 583–588, 2008. http://openaccess.city.ac.uk/2109/.

[5] Liang, Dawen, Haijie Gu, and Brendan O'Connor. *Music Genre Classification with the Million Song Dataset.* Machine Learning Department, CMU, 2011. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.700.2701&rep=rep1&type=pdf.

[6] Laurier, Cyril, Jens Grivolla, and Perfecto Herrera. *Multimodal Music Mood Classification Using Audio and Lyrics.* In Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on, 688–693. IEEE, 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4725050.

[7] Schindler, Alexander, and Andreas Rauber. *Capturing the Temporal Domain in Echonest Features for Improved Classification Effectiveness.* In International Workshop on Adaptive Multimedia Retrieval, 214–227. Springer, 2012. http://link.springer.com/chapter/10.1007/978-3-319-12093-5_13.

[8] West, Kristopher, and Stephen Cox. *Features and Classifiers for the Automatic Classification of Musical Audio Signals.* In ISMIR. Citeseer, 2004. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.443.5612&rep=rep1&type=pdf.

[9] Padial, Jose, and Ashish Goel. *Music Mood Classification.* Accessed December 16, 2016. http://cs229.stanford.edu/proj2011/GoelPadial-MusicMoodClassification.pdf.

[10] Bergstra, James, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. *Aggregate Features and ADABOOST for Music Classification.* Machine Learning 65, no. 2–3 (December 2006): 473–84. doi:10.1007/s10994-006-9019-7.

[11] Corona, Humberto, and Michael P. O'Mahony. *An Exploration of Mood Classification in the Million Songs Dataset.* In 12th Sound and Music Computing Conference, Maynooth University, Ireland, 26 July-1 August 2015. Music Technology Research Group, Department of Computer Science, Maynooth University, 2015. http://researchrepository.ucd.ie/handle/10197/7234.

[12] Dolhansky, Brian. *Musical Ensemble Classification Using Universal Background Model Adaptation and the Million Song Dataset.* Citeseer, 2012. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.4162&rep=rep1&type=pdf.

[13] Tristan Jehan, David DesRoches, *Echo Nest API: Analyze Documentation*, http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf

[14] Ellis, Daniel PW. *Classifying Music Audio with Timbral and Chroma Features*, In ISMIR, 7:339–340, 2007. https://www.ee.columbia.edu/~dpwe/pubs/Ellis07-timbrechroma.pdf.

[15] Juan Pablo Bello, *Low level features and timbre*, New York University, http://www.nyu.edu/classes/bello/MIR_files/timbre.pdf