Era Classification of Songs using the Million Song Dataset

CS 689 Project Report

Siddharth Udayappan Chidambaram
Goutham Devarayasamudram

Introduction

The purpose of our project is to predict which era a song might have come from. The last 50 years have seen the rise and fall of a variety of music genres and tremendous diversity in the kind of music being played. Being Rock music enthusiasts, we were curious to know how songs have evolved over the years and whether there is any pattern in their variations. We were aware of the difficulties of such a task. After all, music has been around for as long as humans can remember. There is no reason why we should expect to identify significant variations in song structure over a span of 50 years. But based on our knowledge of music over the years, we intuitively felt that the music scene has indeed changed tremendously and wanted to put this intuition to test. Era Prediction for songs is not a problem that has been studied extensively before, probably due to the lack of large datasets. Era prediction also has uses other than just satiating our curiosity. Many users tend to prefer music from a particular time (We ourselves swear by the Hard Rock of the 1970s). With Era Prediction, it is possible we can serve better music recommendations by letting people know of bands from the user's preferred musical era.

Dataset

The Million Song Dataset (MSD) was used for the project. The MSD has audio, lyrical data for over a million songs. Most of the songs have the year of their release which we use as labels. The dataset is a little skewed in that the representation of all years is not uniform. There are way more songs from the 1990s and 2000s than the 60s or 70s. Hence, even though the MSD has a million songs, we ended up using 60000 songs to ensure that all decades/eras are equally represented.

Every song in the Million song dataset is represented by 54 attributes of which some are song level attributes represented by one number and others like the timbre, pitch are segment level attributes and hence are represented as vectors.

3. Approach

3.1 Data Preprocessing

The first step in the entire process was to figure out the different eras into which we wanted to partition the songs. We considered various alternatives. We initially started off with decades and the decades we were interested in are from 1960 to 2010. We did not consider the years before 1960 since the MSD did not have enough songs in this timeframe. We also experimented with larger timeframes. The one that has given us the best accuracy so far is trying to classify songs into 3 eras

- prior to 1975
- between 1975 and 1995
- After 1995

One reason for selecting these timelines was because these yielded the best accuracy. Another reason is the fact that the late 1960s and the early 1970s are often considered a pivotal time in rock music history when there was a considerable transition in the kind of music being played.

3.2 Feature Extraction

Feature extraction was the most challenging task in the entire process. After experimenting with a series of features we finalized on using the loudness, duration, pitch and the timbre variations across the duration of the song.

The 2 Figures in (1) clearly explain the increase in average loudness levels through the years and the average duration of the songs seems to follow a similar pattern as loudness although it hits a plateau after a point.

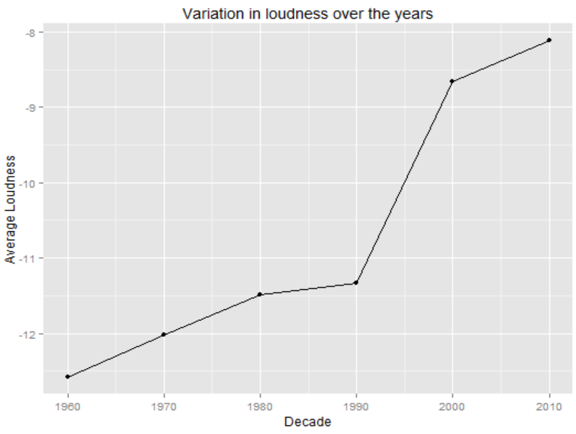

Figure 1(a)

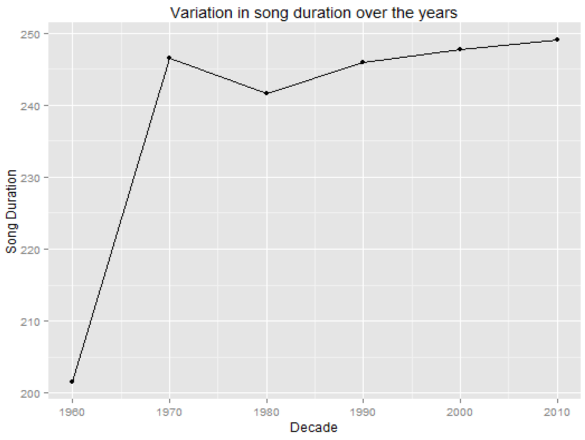

Figure 1 (b)

The timbre and the pitch data however cannot be incorporated as features directly as they are not represented as a single number and vary for each song based on the duration of the song. For instance, the timbre value of each segment is represented by a 12 dimensional vector were these numbers represent the principal components of Mel-frequency cepstral coefficients (MFCCs).

We tried out a number of methods to get a constant number of timbre and pitch features for every song. Some preliminary methods include choosing the most effective (largest) MFCC value for each segment and counting the number of times every component dominates in a song. This method yielded about 50% accuracy and hence we decided to try more effective methods.

Method 1: Using covariance and mean of each component.

The timbre and the pitch varied with the duration of the song and hence we decided to find the average and the covariance of these features across every song. The average resulted in 12 features representing the average across each dimension. Since the covariance matrix is symmetric we decided to choose just the upper triangular elements of this matrix resulting in 78 features. This yielded 90 values each for pitch and timbre. These features along with the loudness and the duration of the song lead to a 182 feature vector for the classification process.

Method 2: Using a clustering based approach

We adopted an approach similar to the method suggested in [2]. Since the number of segments in every song varies there is a need to combine them appropriately to obtain fixed number of features. Hence a new representation is adopted. All the segments from the songs across all the eras are collected and clustered using K-means. The number of clusters was chosen on a trial and error basis and we observed that the best performance was obtained with 30 clusters. Once the segments are clustered, we count the number of segments which fall into each of the 30 clusters and get 30 counts for each song. This along with the other features like loudness and duration thus gives a 32 dimensional feature vector to the supervised classifier. However this method does not give a better accuracy than the average and covariance based method.

3.3 Classification

We made use of standard supervised classification algorithms. The classifiers we tried were:

1. Support Vector machines: After trying out the linear SVM and a set of polynomial SVMS we observed that the Radial Basis function based SVM performed the best and gave excellent accuracies.
2. Nearest neighbors algorithm: The nearest neighbors algorithm was used as the baseline algorithm its simplicity and the ability to scale to larger datasets makes it a very viable option. It gave the lowest accuracy amongst all the classifiers we tried.

3. Logistic Regression: This gave consistently decent accuracies but was easily the slowest of all the classifiers we tried.
4. Gradient boosting: Gradiant Boosting gave the best accuracies without scaling which was slightly less than what we got with SVM.

4. Experimental Results and Analysis

We used Python and sklearn for implementing these algorithms. The process included the data set of 60000 samples (20000 samples in each era). 80% of the samples were used for training and 20% was used in testing. We also performed 5-fold cross validation to make sure that the results were consistent

(For 30000 training samples)	Method 1	Method 2
SVM (radial basis function)	67.10%	60.91%
Nearest neighbors	52.70%	54.25%
Logistic Regression	65.75%	57.41%
Gradiant Boosting	67.71%	60.46%

Table (1): Accuracy of the Various Classifier for both the methods

The best accuracies were obtained using Method-1 with SVM (RBF kernel) as the classifier. The best accuracy we've got so far is around 68.9% with 16800 songs from each of the 3 eras (Prior to 1975, Between 1975 & 1995 and after 1995). Gradient Boosting and logistic regression also gave consistently decent accuracies at around 65-67%. K-Nearest Neighbors was the weakest of the classifiers with accuracies around 50%. The accuracy increased with an increase in the size of the training data for SVM. The rest of the classifiers seem to hit a plateau at around the 68% mark.

The Figure (2) shows the variation of accuracy with the amount of data. It is very clear the SVM based approach improves with the increase in the amount of data. In the case of Gradiant Boosting and the logistic regression based methods the accuracy initially increases with the increase in data, however after a point (around 45000 data points) the accuracy drops. This can be attributed to over fitting. The nearest neighbor approach does not perform well on comparison with the other methods but it increases with the increase in the amount of data as well.

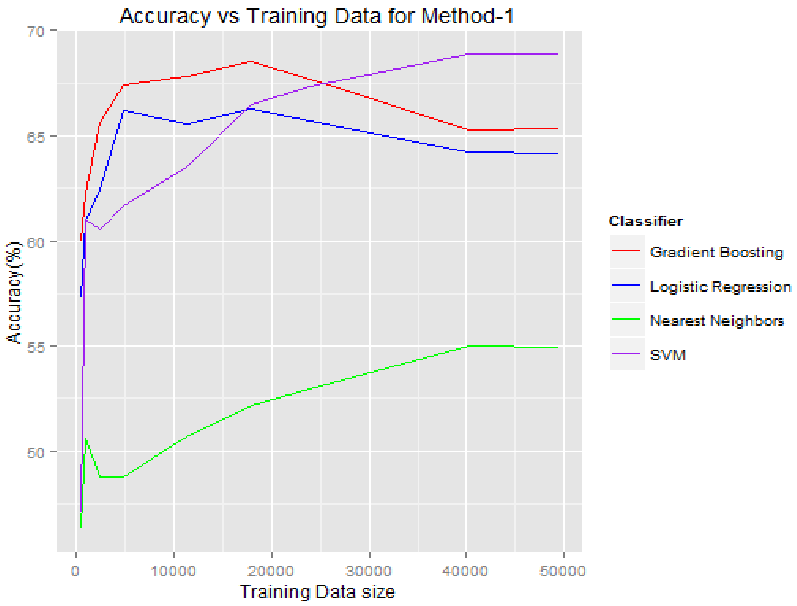

Figure 2: Variation of accuracy with training data size

One other interesting observation is the drastic difference in accuracies for the SVM classifier with and without scaling. The scaling results in the design matrix having zero mean and unit variance. SVM without scaling gave an accuracy of around 33% (as bad as random) whereas after scaling, it gave close to 70% accuracy. Interestingly, both logistic regression and Gradient boosting gave around 65%-68% accuracy without scaling and after scaling, the accuracies were around 63-65%.In case of logistic regression and gradiant boosting the scaling actually resulted in less accuracy (by around 3-4 %) though scaling did result in a significant decrease in the time taken by Logistic Regression to finish classification.

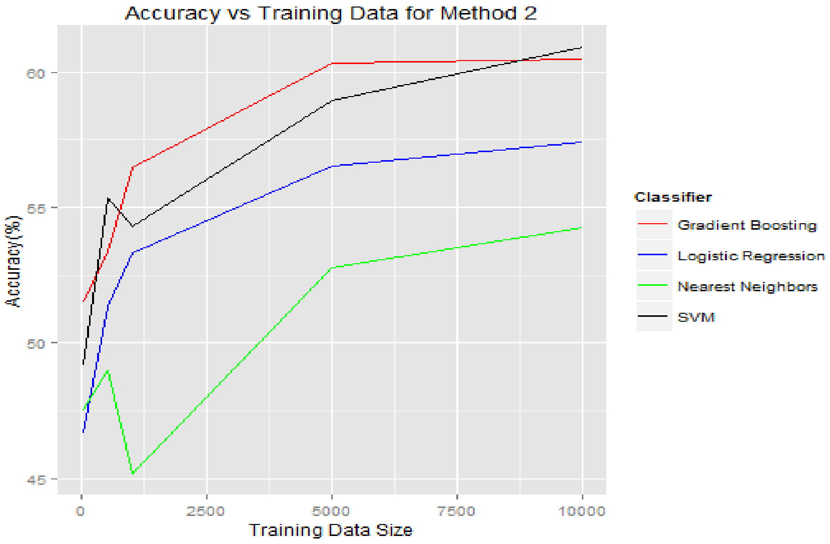

Figure 3 : Variation of accuracy with training data size

Figure (3) shows the variation of accuracy with the size of the training data. We constrained ourselves to training with 30000 samples for this method. All the four algorithms gave accuracies in the range of 55-60% with SVM being the best. Gradiant Boosting and Logistic Regression seemed to show no change after the 7000 sample limit unlike SVM which improved with increase in data size.

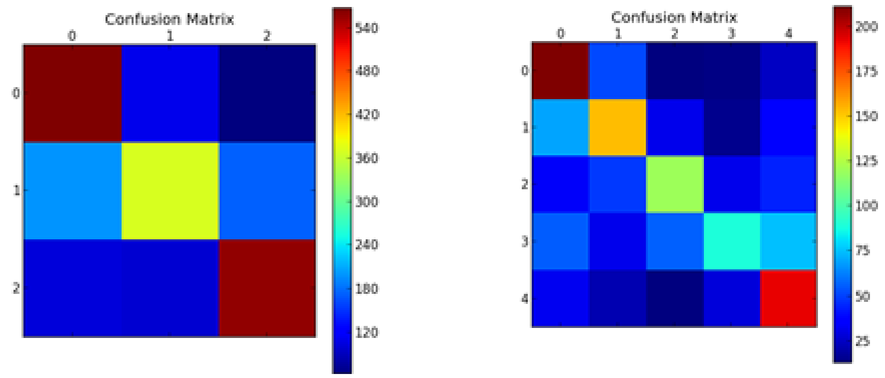

Figure 4(a)

Figure 4(b)

Figure 4(a) : Confusion matrix for era classification
Figure 4(b) : Confusion matrix for decade classification

In Figure 4(a), the labels 0, 1 and 2 correspond to the eras prior to 1975, between 1975 & 1995 and after 1995 respectively. Not surprisingly, the most misclassified era was the middle one, between 1975 and 1995 with a little more than 50% accuracy. Out of a total of 747 songs from that era, 200 were misclassified as belonging to before 1975 and 175 were classified as belonging to after 1995. Both Prior to 1975 and after 1995 have nearly an equal share in the misclassified songs from the middle era. On the other hand, we got an accuracy of 76.5% for predicting songs belonging to prior to 1975 and an accuracy of 73.4 for songs after 1995.

Using the 5 decades from 1960 to 2000 as the eras, we got an accuracy of around 50%. Figure 4(b) shows the confusion matrix for the decade classification with 1960-1970 being represented as 1 and 1990-2000 being represented as 5. The confusion matrix for the decade classification shows that it follows a similar trend as the Era classification.

The gradual increase in average loudness of music over the years was an interesting statistic and we tried classifying based on the loudness alone. This gave us an accuracy of only around 50% and we didn't pursue this further.

Conclusion

While we are quite satisfied with the accuracies we've got so far, we were expecting to see much greater variations in features and hence, greater accuracies. The timeframe we considered for classifying the songs is somewhat broad, being 15 year intervals. We would have loved to see good accuracies for classifying over a narrower period say 5 years or less. Our initial motivation for pursuing this project was to analyze variations in lyrical themes over the years and we tried classifying based on the lyrics alone but that proved to be infeasible. Era Prediction is not a task that is often tackled by the Music Information Retrieval Community and we are glad to have contributed something in this direction.

Acknowledgements

We are thankful to Prof. Sridhar Mahadevan for giving us the opportunity to work on this project. We've always been great fans of Rock Music and thought it might be interesting to apply our recently learned Machine Learning techniques and see if we can uncover any fascinating patterns. Not only did we find fascinating stuff about Music but over the course of the project, we learnt a lot about what it’s like to apply Machine Learning on real world datasets. The time spent in doing the project only convinced us how vast the subject is and how much there is still for us to learn.

References

[1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.
[2]Measuring the evolution of contemporary western popular music http://www.nature.com/srep/2012/120726/srep00521/full/srep00521.html
[3]A Learning-Based Model for Musical Data Representation Using Histograms
[4] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011