CS419M: Project Report
Genre Identification

Sucheta Ravikanti, 160040100
Arunabh Ghosh, 150070006
Dimple Kochar, 16D070010

November 3, 2018

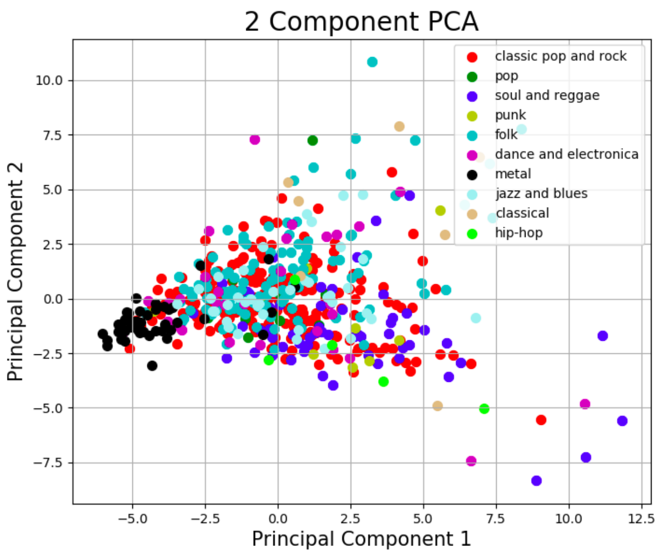
1 Aim of the Project

Identifying what genre a particular song belongs to has been a cakewalk for humans. Can we train the machines do this job for us? With this motivation in mind, we used Machine Learning as a tool for implementing this task of genre identification.
We will briefly take you through our escapades in this report. We will discuss the methods we used for exploratory data analysis, feature selection, hyperparameter optimization, and eventual implementation of several algorithms for classification.

2 The Dataset and Features

For this project, we used the Million Song Genre Dataset, made available by LabROSA at Columbia University. In this dataset, each training example is a track corresponding one song, one release, and one artist. Among the many features that this dataset contains for each track is genre, which was of obvious importance to our project as it is the ground truth which we aimed to predict with our models.

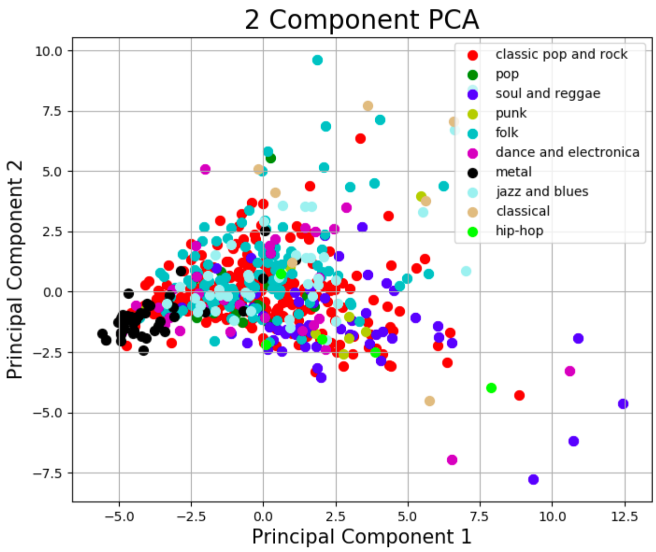
Our first step would be to visualize this data, see how well it can be clustered and analyze if the addition of extra features can improve the clustering of the data or not. We first normalized and scaled the original dataset, and then plotted the data using PCA. Here's how it looked.



Research shows that when a person is listening to a piece of music, the brains attention is heightened during transitions, which frequently coincide with the loudest parts of the music. This indicates that the loudest part of the song might have important information regarding the genre of the dataset. So, we decided to include the timbre feature corresponding to the loudest parts of the song.

Timbre is that quality of the sound which allows our ear to distinguish two sounds which have the same pitch and loudness, mainly determined by the harmonic content and envelope of the sound. The sound of each instrument is each instrument is characterized by its timbre and its an important feature in the classification of genres as many genres are classified by the instruments used in them. For example a saxophones are typically used in Jazz.

Here's how the clusters look after we include this new features, and it does seem that it has slightly better clustering than the previous plot and our faith is reiterated after we calculate the Index of dispersion and see that it falls slightly in the latter case. The index of dispersion basically tells us how dispersed (less clustered) is our dataset by calculating the variance of each cluster divided by the distance between means.



3 Classification of Genres

After completing the exploratory data analysis, and including all the relevant features we now turn our attention to supervised learning methods like Random Forests. For selecting the feature we should use for splitting at each node we used the Gini index G, which is often thought of as a measure of region purity:

$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}) \tag{1}$$

where K is the number of classes and p_{mk} is the proportion of training observations belonging to the kth class in the mth region.

While one single classification tree might not have extraordinary predictive accuracy in comparison to other machine learning methods, aggregation of decision trees can substantially improve performance. This is why we decided to use **Random forests**.

We first used Bayesian optimization, which is explained in greater detail in section 4, to tune several hyperparameters. The results of this optimization indicated that the optimal minimum number of samples required to split an internal node in a tree is 2, the optimal minimum number of samples that must be in a newly created leaf is 1, and the optimal number of trees in the forest is 300. Having tuned these parameters, we built a random forest classifier using our training set, which we then used to make genre predictions for our test set.

Using these optimal set of hyperparameters our random forest was able to achieve a accuracy of 56% and a F1 score of 50.65%.

4 Tuning the hyper parameters

Choosing the right parameters for a machine learning model is almost more of an art than a science. Of course for tuning hyperparameters we can do something as simple as random search. However, when you are training sophisticated models on large data sets, it can sometimes take on the order of hours, or maybe even days, to get a single sample from . In those cases, can we do any better than random search? It seems that we should be able to use past samples of, to determine for which values of we are going to sample next.

And here comes **Bayesian Optimization**. Bayesian optimization falls in a class of optimization algorithms called sequential model-based optimization (SMBO) algorithms. These algorithms use previous observations of the loss , to determine the next (optimal) point to sample for. The algorithm can roughly be outlined as follows.

- Using previously evaluated points, compute a posterior expectation of what the loss looks like.
- Sample the loss at a new point, that maximizes some utility of the expectation of. The utility specifies which regions of the domain of are optimal to sample from.

To compute the posterior probability we need a prior on the distribution of loss.

4.1 Gaussian Processes

As the prior to our Bayesian optimization problem, we used a Gaussian Process (GP). Gaussian Processes are defined by a multivariate Gaussian distribution over a finite set of points. What makes Gaussian processes useful for Bayesian optimization is that we can compute the marginal and conditional distributions in closed form.

4.2 Acquisition functions

To find the best point to sample next from, we will choose the point that maximizes an acquisition function. This is a function of the posterior distribution over , that describes the utility for all values of the hyperparameters. The values with the highest utility, will be the values for which we compute the loss next.

What does an acquisition function look like? There are multiple proposed acquisition functions in the literature, but the expected improvement (EI) function seems to be a popular one. With a Gaussian process, this has a closed form:

$$a(x) = E(max\{0, f_{t+1}(x) - f_t(x^*)\}) = \sigma(-u\phi(-u) + \phi(u)) \tag{2}$$

4.3 Bayesian Optimization of Algorithms

We used Bayesian Optimization to optimize three hyperparameters in our algorithm:

- Number of trees in the random forest
- Minimum number of samples required to split an internal node in a tree
- Minimum number of samples that must be in a newly created leaf

The results of this optimization indicated that the optimal minimum number of samples required to split an internal node in a tree is 12, the optimal minimum number of samples that must be in a newly created leaf is 1, and the optimal number of trees in the forest is 196.

5 Results

In this section, we analyze how our primary models per- formed, and discuss reasons why they did not perform even better. First here is how our Random forests with its optimal set of hyperparameters performed on all the genres.

Table 1: Summary of the results

Class	**Precision**	**Recall**	**F1-Score**
Classic pop and rock	0.53	0.90	0.67
Folk	1.00	0.17	0.29
Dance and electronica	0.55	0.47	0.51
Classical	1.00	0.88	0.93
Hip-Hop	0.27	0.23	0.25
Soul and reggae	0.50	0.44	0.47
Punk	1.00	0.17	0.29
Metal	0.70	0.45	0.55
Jazz and blues	0.37	0.69	0.48

Overall Accuracy: 56%
F1-Score: 0.5065

After the training and testing phases, we observed that some genres got clubbed with other genres. This mainly occurs with two genres Folk and Punk as we see in this cases the recall is very low.

Punk was often confused with classic pop and rock, due to the fact that the Punk sample size was relatively small. Additionally, these genres tend to have similar time signatures and make heavy use the electric guitar.

Classical pop and folk were confused. This makes sense, as both have similar time signatures and generally use acoustic instruments. They also have similar loudness, which makes differentiation even more difficult.

It seems that classical did the best in both precision and recall, which makes sense because it generally has a very different timbre than most other genres. The instruments used in classical such as trumpets are unique to the genre.

6 References

1. From Classical To Hip-Hop: Can Machines Learn Genres? By Aaron Kravitz, Eliza Lupone, Ryan Diaz, Stanford University

2. A Tutorial on Bayesian Optimization for Machine Learning: Harvard University

3. A Tutorial on Principal Component Analysis: Jonathon Shlens, Google Research

7 Code

The code we developed in this project has been made open source and can be found in the following link:

https://github.com/Arunabh98/Genre-Identification

All our codes for the random forest classifier, PCA and the Bayesian optimization can be found in the **code** subdirectory. The results of PCA have been stored in the **images** subdirectory. In the research subdirectory you may find the Stanford paper we referred to in the course of our project.