Music Genre Identification

- Rajat Gupta

Abstract: Machine learning algorithms allow us train data based on certain features and predict outcomes. They can be used for classification and regression tasks. This problem set deals with the application of such algorithms in identification of different music genres and artists given a 5 second audio clip. Through the application of Singular Value Decomposition (SVD), dominant features were determined in the audio files and machine learning algorithms were applied to these features along with cross validation to determine their accuracy and errors (through plotting the confusion matrix).

I. Introduction and Overview

Machine Learning algorithms allow us to identify patterns in the data that help us in classification and prediction based on a set of values provided. There are various such algorithms available at our disposal and have been tested here, we have ultimately selected the one with best performance (highest accuracy score).

In this project, I am attempting to answer 3 questions – Can our algorithm classify audio clips of bands belonging to different genres, how accurately will the algorithm work when the same task has to be performed for bands within the same genre and how well does it work when we just want to predict the genre given a 5 second audio clip.

The data is aggregated from the songs downloaded by me, contains 24 songs in each of the categories. The songs are sampled multiple times at different starting intervals to construct the data matrix and as the algorithm remains the same for each of the cases, only the data is modified for the analysis in each case. Cross-validation has also been performed in each of the cases to determine whether overfitting is taking place.

II. Theoretical Background

Most audio samples are stereo – have distinct left and right channels. For our analysis, we will want our audio to be mono – have the same output in left and right channels. Thus, we only keep the left channel for our analysis.

Before we apply machine learning algorithms on the data, we need to identify the features that the algorithm will train on. For this purpose, we use Singular Value Decomposition on the spectrogram of the audio clips to identify the dominant modes associated with an artist. After selection of the dominant features and construction of our data matrix, we apply the machine learning algorithm of our choice.

Machine learning algorithms are broadly classified into 2 categories – supervised learning algorithms and unsupervised learning algorithms. Supervised learning algorithms maps an input to output based on example input, output pairs provided during training – we are providing 'labels' for the data during training. Whereas, for unsupervised learning no labels are provided during training.

In this problem set, we are performing supervised machine learning as we are providing labels for our data – the genre/artist. Different supervised machine learning algorithms used to answer the questions mentioned in section 1 here are explained below:

1. Naïve Bayes – Based on Bayes theorem, "naive" because it assumes that all the features provided are independent of each other. It is highly scalable but because of the naïve assumption of independence between features, accuracy isn't always high.
2. Decision Trees – Uses Classification and Regression Tree (CART) algorithm, to split the data based on certain features, to create a tree like structure. Decision Trees tend to usually over fit data, thus it is required to add regularization to limit this overfitting.
3. Support Vector Machines (SVM) – This algorithm tries to fit the widest possible "street" between data belonging to different classes. It is extremely powerful and can be used for linear and non-linear classification but requires scaling to be performed prior to training of the algorithm.

It is also to be noted that cross validation has been performed for each of the cases to determine whether overfitting is taking place.

We calculate accuracy and plot the confusion matrix after testing each algorithm. The confusion matrix is a table that tells us about the performance of our algorithm – it tells us how many instances for each class were misidentified.

III. Algorithm Development and Implementation

This problem set was solved using python, version 3.6.

The steps followed to solve the problem are summarized below:

1. We load the data from the .mp3 files and convert them to .wav format, keep only the left channel (convert to mono) and extract different 5 second clips from a song. The last part is performed by the lines:

   ```
   for i in range(0,20):
           song_data.append(data[5*i*samplerate:5*(i+1)*samplerate,0])
   ```

   Here, we are sampling each song 20 times. Note that [.., 0] indicates that we are storing only the left channel data. Sample rate and data are the sampling rate and audio data for the particular audio file and are extracted through the command:

   ```
   samplerate, data = scipy.io.wavfile.read(wav_file)
   ```

2. The variable 'song_data' contains all the audio data of interest to us. Using this variable we will construct the spectrograms for each of the audio clips and store the values in our data matrix X. This is done as follows:

```
for s in range (len(song_data)):
  k = 0
  spec,_,_,_ = plt.specgram(song_data[s], Fs = sample_rate[s])
  rows, cols = spec.shape
  for i in range (rows):
            for j in range (cols):
                      a[k] = spec[i][j]
                      k += 1
            X[:,s] = a
```

Here, for each song we are constructing a spectrogram and extracting the data from the spectrogram, reshaping it to a column and adding it to our data matrix 'X.'

3. We then perform the SVD on matrix X using the command –

```
U, sigma, V = np.linalg.svd(X), full_matrices=False)
```

The full_matrices = False, signifies that reduced SVD has to be performed.

4. Once we plot the Singular Value Spectrum and identify the number of features we need for our analysis, we will split the data into training and testing sections and assign labels to each of them. As the data that is in the training and testing set is to be selected at random, we use the following command:

```
x1 = np.random.permutation(480)
```

This generates a random shuffling of values ranging from 0-479. 480 has been passed because, there are 24 songs in each category and each of these songs have been sampled 20 times, making each category have 480 audio clips.

5. After splitting the data into training and testing tests, assigning labels to each of them, we apply our machine learning algorithm. The following lines denote application of SVM:

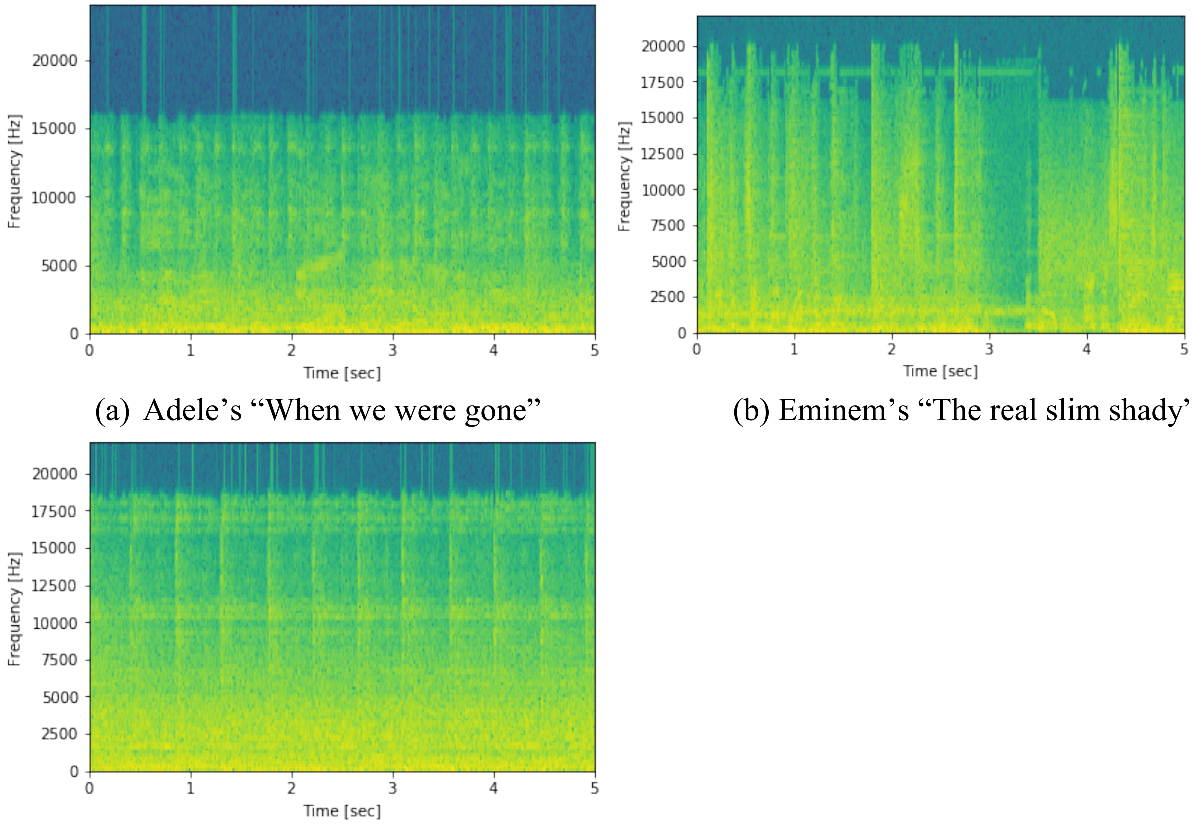
```
pipeline = Pipeline((
        ("scaler", StandardScaler()),
        ("svm_clf", SVC(kernel='rbf', C=21, gamma = .00095) )))
pipeline.fit(xtrain, ytrain)
```

We are first scaling our features before application of the SVM algorithm. Kernel, C and gamma are parameters that were tweaked to get the best performance from the algorithm.

6. We then make predictions using this algorithm on our test data set and evaluate the performance of this algorithm by calculating the accuracy and plotting the confusion matrix.
7. The steps 4-6 are repeated so as to perform cross-validation for each algorithm
8. The data is then changed in order to answer the other questions mentioned in Section 1.

IV. Computational Results

It is seen that music belonging to different genres have different spectrograms as seen in Image 1 where 3 pieces of music are presented. Our training set has 450 audio clips of each genre = 1350 audio clips and our testing set has 30 audio clips of each genre = 90 audio clips. Each audio clip is 5 seconds in length.



(a) Adele's "When we were gone"

(b) Eminem's "The real slim shady"

(c) Oasis' "Champagne Supernova"

Figure 1: Spectrogram of songs for artists of different genres

After running SVD on the dataset and plotting the singular value spectrum, we select 190 features among 1440 features available to us as most of the variance is contained in these modes as shown in figure 2.

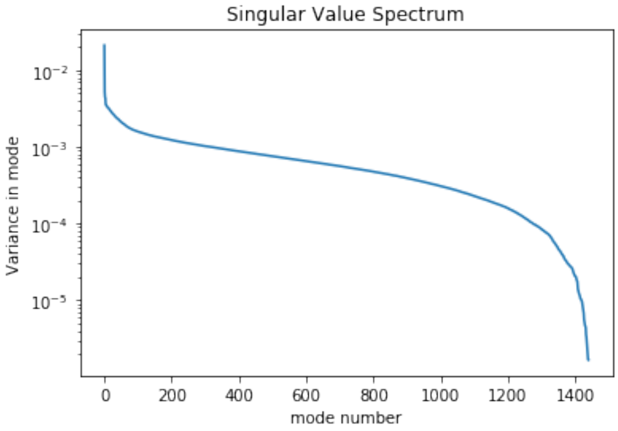
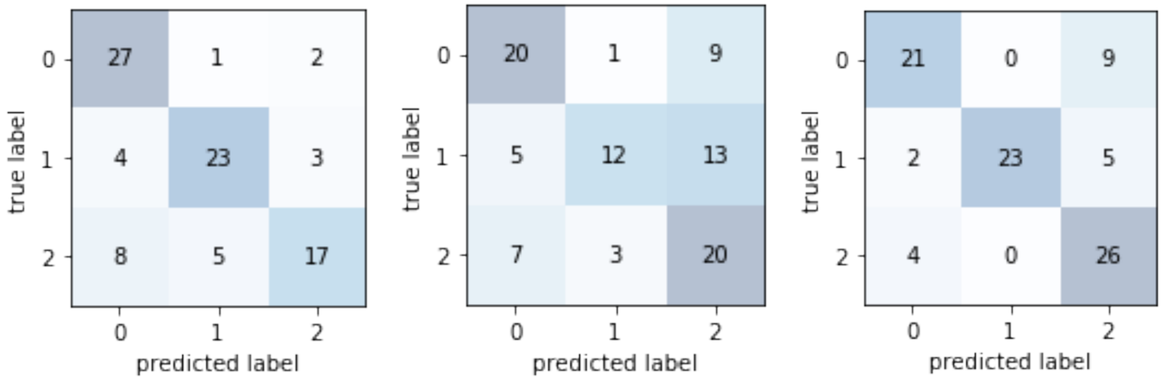


Figure 2: Singular Value Spectrum for the matrix X

Different algorithms were used when answering question 1: Can the machine learning algorithm classify audio clips of bands belonging to different genres. It was seen that SVM gave the best accuracy score (~75%) among itself, decision trees (~60 %) and naïve bayes (~45%) classification algorithms. The confusion matrix achieved for SVM algorithm in this case can be seen in figure 3(a).



(a) Case 1 (b) Case 2 (c) Case 3

Figure 3: Confusion Matrix for SVM algorithm for 3 cases being tested as mentioned in Section 1

The comparison in case (a) is being performed between artists – Adele (label 1), Eminem (label 2) and Oasis (label 3) each belonging to a different music genre – Pop, Hip-pop and Rock respectively. As seen from the confusion matrix, the SVM algorithm classifies most of the audio clips to be music sung by Adele, making 39 predictions for this artist out of which 27 were correct (in total there were 30 songs sung by Adele in the test set), whereas it classifies only 22 audio clips to be sung by Oasis out of which only 17 were correct.

While performing cross-validation the distribution of correct values changes along with the accuracy score but, SVM on an average gave the best performance among the 3 algorithms tested. Thus, for case 2 and 3, the SVM algorithm with the same parameters, number of features as in case 1 but with different data was used.

For case 2 we are attempting to answer the question: how accurately does the algorithm work when the classification has to be performed for bands within the same genre. Running this algorithm and performing cross-validation we find that the accuracy score drops to ~60%. This is expected because within the same genre there is less difference between the audio clips and thus their respective spectrograms are relative to case 1 more similar and as we are using these spectrograms for our analysis, the accuracy score drops. From Figure 3 (b), we can see that the algorithm classifies most of the audio clips as songs by Oasis (label 1) and U2 (label 3), less than half are identified as songs by Coldplay (label 2).

For case 3 we are interested in answering the question: how well does the algorithm work when we just want to predict the genre given a 5 second audio clip. Here label 1, 2 and 3 denote Pop, Hip-Pop and Rock respectively. It is seen that the accuracy score increases here to around 79%. This could be due to the fact that incorporating songs from different artists in the data set for each of the genres makes the data set a little diverse and helps in identifying a lot of ‘edge cases’ which might have been left out in case 1. It is also seen here that the distribution of predictions between the different genres is more uniform between the different classes and most of the predictions are made for the genre Rock (class 3) and least for Pop (class 1).

We also see that the accuracy score varies with the number of features we select before training the algorithm, therefore we need to limit the features to the ones which capture the most amount of variance.

V. Summary and Conclusion

Thus, we have been able to identify songs belonging to different artists, genre using machine learning techniques. We can say that the accuracy of the machine learning algorithm which we are utilizing depends on the dataset – more number of training points we have, better will be the accuracy which we actually get.

We have successfully used Singular Value Decomposition to extract the dominant features in the dataset and have performed training and classification using machine learning techniques. It is seen that SVM gives a much better accuracy score than Naïve Bayes and Decision Tree Classifiers. Also, the algorithm works much better for the cases when we are trying to classify artists in different genres and music in different genres. When we perform classification for artists within the same genre, the accuracy drops considerably as the data i.e. the audio clips and the spectrogram are much more similar in this case.

Appendix A

Python 3.6 used for analysis.

Libraries used in the program:

- Numpy – Python package for scientific computing.
- Matplotlib – Library useful for making different plots and figures.
- Scipy.io.wavfile – Library used for reading .wav files
- Pydub – Python package to read audio files (.mp3)
- Sklearn – Machine Learning library for Python.
- Glob – Python package used for traversing libraries and analyzing files.

Functions used in the program:

- AudioSegment.from_mp3(song) – Extract information from mp3 file
- sound.export(wav_file, format='wav') – Export the mp3 song to a wav file
- samplerate, data = scipy.io.wavfile.read(wav_file) – Read the .wav file and extract the data and sample rate.
- plt.specgram(song_data, Fs = sample_rate) – Create the spectrogram from the audio data and the sample rate. Returns the spectrogram, time, frequency and image.
- np.linalg.svd(X, full_matrices=False) - – Performing the SVD on the input matrix 'mat'. The flag, full_matrix = False signifies that reduced SVD is required to be performed.
- np.random.permutation(480) – Generates a random sequence of numbers in the range of 0-479
- plt.semilogy(x,y) – Plot the y values on a log scale.
- Pipeline((("scaler", StandardScaler()),("svm_clf", SVC(kernel='rbf', C=21, gamma = .00095)))) – Creating a pipeline which performs feature scaling using the function StandardScaler() and initializes the SVM algorithm with parameters, kernel, C and gamma.
- pipeline.fit(xtrain, ytrain) – Train the algorithm based on the data and labels provided.
- pipeline.predict(xtest) – Predict the labels based on the algorithm trained above and the data provided in xtest.
- accuracy_score(ytest, ypred_svm) – returns the accuracy score based on the actuall results that should be obtained i.e. ytest and the ones predicted by the algorithm ypred_svm.
- confusion_matrix(ytest, ypred_svm) – generate the confusion matrix based on the actual and predicted labels.
- plot_confusion_matrix(conf_mat=conf_svm) – Plot the confusion matrix in a manner which is visually understandable.