Automatic Speech Recognition

CS 753

Music Genre Classification

Kalpesh Patil (130040019)
Rishabh Raj (130050045)
Chandrakanth BKC (130050053)

Guide: Prof. Preethi Jyothi

May 31, 2017

Abstract

Music genre classification is one of the most interesting topic of research in the domain of Music Information Retrieval (MIR). It finds applications in the real world in various fields like automatic tagging of unknown piece of music (useful for apps like Saavan, Wynk etc.), personalized music experience (Pandora, which generates images according to genre) etc. In this project we have explored various techniques to carry out this task. We have tried out methods like Vector Quantization, GPPS-NN, GPPS-SVM, DNN, CNN etc. We will briefly explain each of these methods and the experimental results that we obtained. In the end we will also try to analyze the results to get more insight into the problem.

1 Task Definition

The task in hand is to predict the genre of a piece of music from the utterance. Following stages are involved here, which will be described in detail later.

- Feature extraction from the uttered sound
- Classifier training

Data

In this project, we are using the GTZAN Dataset which has a collection of total 1000 music clips of 30 seconds duration, 100 songs of each genre. We decided to go ahead with only 4 genres namely, Classical, Metal, Pop and Jazz similar to [3] We pre-processed the audio files to convert them to wav file format from au for further extraction of MFCC features.

We have used 80-20 train test split to demonstrate results. We experimented with several different methods. Details and results of each of these methods is explained in the next section.

Cluster size	100	200	400	600	800
Accuracy	93.75	97.10	96.25	95.00	93.75
Precision	94.41	97.61	96.53	95.54	94.64
Recall	93.75	97.39	96.35	95.31	94.27
F-score	93.80	97.44	96.30	95.17	94.03

Table 1: Performance of VQ based approach with different number of clusters

2 Methods and Results

2.1 Vector Quantization

Method Description

The Vector Quantization method involves generation of a codebooks for all the genres, of certain size. The codebook is generated by from the training data by clustering the feature vectors of all frames of all the training music samples for a given genre. Generally,a variant of K-Means, *MiniBatchKmeans* is used due to its computational effectiveness against normal K-Means algorithm for large number of data points. For classifying a test song, we compute euclidean distance of each frame of a test song from the centroids. The minimum distance from all centroids of a particular genre is computed and is averaged across all frames of the given song clip. The test frame is predicted to belong to the genre which minimizes this total average distance.

Results and Discussion

Table 1 shows results of this technique by varying size of the codebook. We can see that the accuracy increases as the number of clusters increase, but only upto a certain point after which it starts decreasing with increasing number of clusters. Apart from the accuracies, we also analyzed other aspects such as confusion matrix, precision, recall and F-score; which gives us more insights into performance rather just accuracy. Figure 1 represents the confusion matrix for optimal parameters. We observe the highest confusion between genre 'jazz' and genre 'classical'.

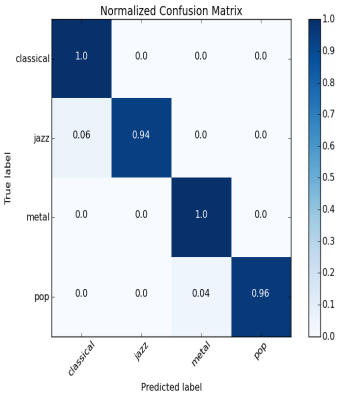

Figure 1: Confusion Matrix for VQ based approach

2.2 GPPS

Method Description

Classical approaches based on *GMM-UBM* setting have been tried for a variety of tasks like accent recognition [2], speaker age estimation[1]. But we didn't find any study which uses GPPS for the task of music genre classification and decided to incorporate *GPPS* in our project. The fundamental reasoning behind using GPPS is to convert a sequence of MFCC data (sequence of frames) for a piece of music to a single feature vector representing that song. The approach can be summarized as follows:

- **GMM-UBM training**
  From the MFCC feature vectors a Universal Background Model *(UBM)* is trained. It is assumed to be mixture of Gaussian distributions *(GMM)*. Number of distributions used for training a UBM is an important parameter here. We have studied results by varying number of distribution components of GMM. Using the standard procedure (EM algorithm), we fit a GMM model on training data and estimate weights, means and covariances of mixture components. Since we have limited amount of data, large number of parameters can't be estimated. Hence we assume diagonal covariance matrices for mixture components.

- **GPPS extraction**

GPPS vector for an utterance can be computed as follows.

$$Pr(o_t|\lambda) = \sum_{j=1}^{J} w_j Pr(o_t|\mu_j, \Sigma_j)$$
$$\lambda = \{w_j, \mu_j, \Sigma_j\}, j = 1, 2..J$$

Here o_t is acoustic feature vector at time t (39 dimensional MFCC in our case). w_j, μ_j, Σ_j are estimated parameters of j^{th} mixture component. $Pr(o_t|\mu_j, \Sigma_j)$ is the probability value according to Gaussian distribution. After computing these values for each frame, we proceed to compute GPPS vector.

$$\kappa_j = \frac{1}{T} \sum_{t=1}^{T} \frac{w_j Pr(o_t|\mu_j, \Sigma_j)}{\sum_{j=1}^{J} w_j Pr(o_t|\mu_j, \Sigma_j)}$$
$$\kappa = [\kappa_1, \kappa_2, ...\kappa_J]$$

Here T is total number of frames in the piece of music and κ is GPPS vector of that music. Thus we extract J dimensional GPPS vector for train and test data. This feature vector is fed to top layer classifier.

- **Classifier**
  Our aim is to train a classifier which takes J dimensional GPPS vector as input and predicts the genre. We can use any off-the-shelf classifier for this purpose. *SVM* and *Neural Network* are the most famous.
  - *SVM*: We fine tuned the parameter C, gamma and used rbf kernel. Following parameters seems to produce good results.

    kernel = rbf, C=100.0, gamma=0.5
  - *Neural Network*: We tried with different architectures of Neural Network for the classification task. Following architecture seems to produce good results.

    InputLayer(J), Dense(10,relu), Dropout(0.5), Dense(4,softmax)

Results and Discussion

We have reported the results for varying J (number of mixture components) in Table 2 for SVM and NN. We obtain improvement in the accuracy as

	NN				SVM			
GMM components	64	128	256	512	64	128	256	512
Accuracy	86.25	87.50	90.00	93.75	91.25	91.25	90.00	87.5
Precision	86.72	88.67	91.04	94.31	92.42	92.39	91.72	90.19
Recall	87.18	87.91	90.73	93.75	91.45	91.97	90.62	88.54
F-score	86.60	87.93	90.52	93.77	91.40	91.63	90.39	88.07

Table 2: Performance of GPPS approach with different number of mixture components

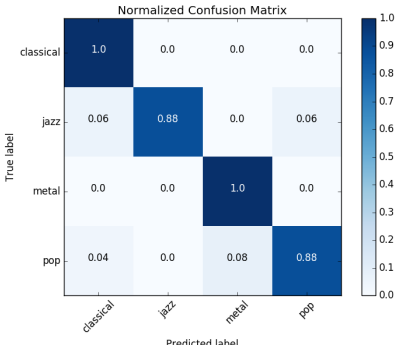

Figure 2: NN

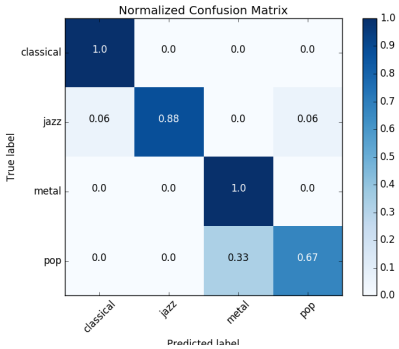

Figure 3: SVM

Figure 4: Confusion matrix for GPPS based technique

number of clusters increase in the case of NN, while no such trend is observed for SVM. Figure 4 represents the confusion matrix for NN and SVM each. We observe the highest confusion between genre 'metal' and genre 'pop', i.e. true label being 'pop' getting predicted as 'metal'.

2.3 DNN

Method Description

Deep Neural Networks have been extensively used in literature for classification tasks with highly non-linear class boundaries. Rather than assuming the frames to be independent of each other, in this method we try to use the context of the frame to provide more information to the classifier. First of

all we create a context input vector for every frame by appending few frames before and after to the original frame and give this as an input to Neural Network. Let W be the window considered on both the sides of the frame in consideration. Therefore input to NN will be $39(2W + 1)$ dimensional. We used W equals to 5. All such contextualized frames across all utterances in training data are gathered together and tagged with respective genre id. This forms training set for the neural network. We train this network on available training data with usual *cross entropy loss*. Once this network is trained, we use it to predict genre id of all frames in the test utterance. explored *maximum likelihood*, where we take sum of log probabilities of each language across frames. Sum of log probabilities corresponds to product of probabilities which is similar to maximum likelihood. Let $Pr(y_n = i|x_n)$ be the probability that n^{th} frame belongs to language i given input x_n. Therefore

$$\hat{L} = \underset{i}{argmax} \prod_{n=1}^{N} Pr(y_n = i|x_n)$$

$$\hat{L} = \underset{i}{argmax} \sum_{n=1}^{N} log(Pr(y_n = i|x_n))$$

$\hat{L}$ is the genre predicted for the given test song. One might argue that $\{y_n\}$ are not independent, hence probability can't be written as product of individual terms. Following architecture was used for DNN

*InputLayer(*429*), Dense(2650,relu), Dropout(0.5), Dense(2650,relu), Dropout(0.5), Dense(2650,relu), Dropout(0.5),Dense(4)*

Results and Discussion

Performance and confusion matrix of DNN based approach are mentioned below. We realize that the performance isn't at par with GPPS or VQ. We agree that we didn't do thorough parameter tuning of network architecture (depth, width etc.) due to computationally expensive and time consuming training of DNN, although we tried our best to find the best architecture. Another reason for such low performance could be lack of enough training data.

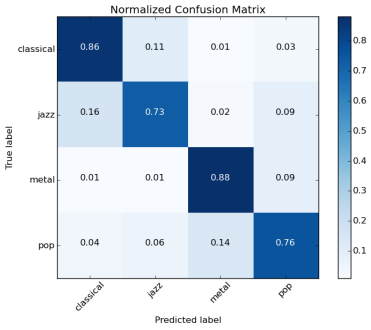

Figure 5: Confusion matrix for DNN based approach

Accuracy	80.72
Precision	80.30
Recall	80.57
F-score	80.36

Table 3: Performance of DNN based approach

2.4 CNN

Method Description

2D-CNN are quite popular for classification of images. Weight sharing and local connectivity are the major advantage of CNNs. Here in case of genre classification we can use 1D-CNN, where local connectivity is across time. We have used following 1D-CNN architecture.

*InputLayer(*2998×39*), Conv1D(64,4,tanh), MaxPool1D(2), Conv1D(128,4,tanh) ,MaxPool1D(2), Conv1D(256,4,tanh),Dense(512),Dense(4)*

Results and Discussion

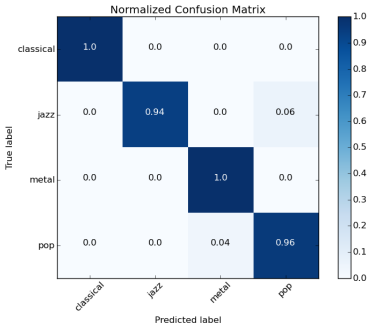

Figure 6: Confusion matrix for CNN based approach

Accuracy	97.50
Precision	97.77
Recall	97.40
F-score	97.54

Table 4: Performance of CNN based approach

3 Discussion

We will like to point out few interesting observations in this section.
Amongst the genres, pop and jazz were most often confused with other genres. Clssical and metal were least confused, which was kind of expected. This is pretty evident from the confusion matrices of different classifiers. Pop and jazz were most often misclassified as follows:

- True label: Pop, predicted label: Metal
- True label: Jazz, predicted label: Metal
- True label: Jazz, predicted label: Classical

We further analyzed the songs which got misclassified and could identify a small section of songs which always got misclassified, regardless of the classifier used. Examples include pop00014.wav, jazz.00000.wav which can be found in the GTZAN Dataset under the folder pop and jazz respectively. To us, pop00014.wav indeed sounded very familiar to jazz music, while jazz.00000.wav being classified as metal was a bit puzzling. However, we guess the fast paced rythm of the song is the reason for it being classified as metal.

4 Conclusion and Future Work

1D-CNN and VQ were found to be the best performing methods amongst the ones which were tried. DNN model using GPPS performed better than DNN model using just the context frames.
A simple extension of the project could be to try the classifiers for the whole GTZAN Dataset comprising of 1000 songs of 10 genres. A possible extension of the project could be to map songs to music bands, given that a band's music generally pertains to the same genre. Another extension could be to use the predicted genre of the song to create an ambience of lighting which matches that genre.

References

[1] M. H. Bahari et al. Speaker age estimation using hidden markov model weight supervectors. In *Information Science, Signal Processing and their*

Applications (ISSPA), 2012 11th International Conference on, pages 517–521. IEEE, 2012.

[2] M. H. Bahari, R. Saeidi, D. Van Leeuwen, et al. Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7344–7348. IEEE, 2013.

[3] M. Haggblade, Y. Hong, and K. Kao. Music genre classification. *Department of Computer Science, Stanford University*, 2011.