CSN-382: Machine Learning Project

Music Genre Classification

Ambar Zaidi	14114009
Saurabh Goyal	14114051
Tanmay Tiwari	14114066

CSN-382
Spring 2017
IIT Roorkee

Data Retrieval Process

Database

- *GTZAN Genre Collection*,
  - 1000 audio tracks
  - Each 30 seconds long
  - 10 genres represented,
  - Each containing 100 tracks.
  - All the tracks are 22050Hz Mono 16-bit audio files in .au format

Data Retrieval Process

Primary data used:

- Categories chosen:
  - rock,
  - jazz,
  - hip-hop, and
  - disco
- Our total data set was 400 songs, of which we used 66% for training and 34% for testing and measuring results.

```
Calcuating MFCC:
        Category: rock
        Category: hiphop
        Category: jazz
        Category: disco
        Category: metal
        Category: pop
        Category: country
        Category: reggae
        Category: classical
        Category: blues
```

Extracting features from database

Feature Extraction

Mel Frequency Cepstral Coefficients (MFCC)

Why use MFCC ?

- a way to concisely represent song waveforms
- time domain waveforms ➔ a few frequency domain coefficients
- Reduced 22050 features to 13 features per frame

Feature Extraction

Mean Matrix , Covariance Matrix and Combination

- Reduced MFCC Matrix of 400x13 to mean matrix of size 1x13
- Reduced MFCC Matrix of 400x13 to Covariance Matrix of size 13x13
- Combined upper half of symmetric covariance matrix flattened with mean matrix resulting 1x104 vector
- Feature tuple = (mean matrix , covariance matrix , flattened matrix , class)

Techniques

4 algorithms

- KL Divergence
- k-Nearest Neighbors (k-NN)
- k-Means
- Multi-Class Support Vector Machine (SVM)
- Convolutional Neural Networks

Kullback-Leibler (KL) Divergence

- We compute distance between two songs via the Kullback-Leibler divergence. Consider ***N0*** and ***N1*** to be the two multivariate Gaussian distributions with mean and covariance corresponding to those derived from the MFCC matrix for each song. Then, we have the following:

$$D_{\mathrm{KL}}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2}\left(\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^{\top}\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right).$$

- However, since KL divergence is not symmetric but the distance should be symmetric, we have used

Distance = KL divergence (p,q) + KL Divergence (q, p)

k-Nearest Neighbors (k-NN)

Used KL divergence for multivariate distribution to find the distance between a song in test set with training set and took class dominant in the nearest k songs.

k-Means

- K-means clustering is a type of unsupervised learning.
- For training, we created clusters equal to the number of categories for classification.
- Then, we mapped each cluster to a category having highest number of instances in that cluster.

Multi-Class Support Vector Machine (SVM)

- **Support vector machines (SVMs)** are a set of supervised learning methods used for classification.
- Used scikit learn implementation for python **sklearn.svm.SVC**.
- Tuned parameters using **GridSearchCV ( )** for optimal model.

Results

Confusion matrix:

	precision	recall	f1-score	support
1	0.92	0.71	0.80	34
2	0.97	0.94	0.96	34
3	0.89	1.00	0.94	34
4	0.87	1.00	0.93	34
avg / total	0.91	0.91	0.91	136

Accuracy: 0.911764705882

Confusion matrix:

	precision	recall	f1-score	support
1	0.50	0.38	0.43	34
2	0.60	0.71	0.65	34
3	0.86	0.53	0.65	34
4	0.48	0.59	0.53	34
5	0.96	0.71	0.81	34
6	0.80	0.82	0.81	34
7	0.63	0.50	0.56	34
8	0.48	0.76	0.59	34
9	0.84	0.94	0.89	34
10	0.78	0.74	0.76	34
avg / total	0.69	0.67	0.67	340

Accuracy: 0.667647058824

Results using kNN(82%):

1. For 4 genres:
   a. Accuracy: 91.2%
   b. F1-score: 91%
2. For 10 genres:
   a. Accuracy: 66.8%
   b. F1-score: 67%

Results

Confusion matrix:

	precision	recall	f1-score	support
1	0.82	0.79	0.81	34
2	1.00	1.00	1.00	34
3	0.89	0.94	0.91	34
4	0.91	0.88	0.90	34
avg / total	0.90	0.90	0.90	136

Accuracy: 0.904411764706

Confusion matrix:

	precision	recall	f1-score	support
1	0.39	0.32	0.35	34
2	0.42	0.47	0.44	34
3	0.51	0.56	0.54	34
4	0.61	0.56	0.58	34
5	0.81	0.65	0.72	34
6	0.74	0.85	0.79	34
7	0.53	0.26	0.35	34
8	0.46	0.56	0.51	34
9	0.85	0.82	0.84	34
10	0.47	0.68	0.55	34
avg / total	0.58	0.57	0.57	340

Accuracy: 0.573529411765

Results using SVM(87%):

1. For 4 genres:
   a. Accuracy: 90.4%
   b. F1-score: 90%
2. For 10 genres:
   a. Accuracy: 58%
   b. F1-score: 57%

Results

Confusion matrix:

	precision	recall	f1-score	support
1	0.69	0.26	0.38	34
2	0.97	0.94	0.96	34
3	0.65	0.97	0.78	34
4	0.74	0.85	0.79	34
avg / total	0.76	0.76	0.73	136

Accuracy: 0.757352941176

Confusion matrix:

	precision	recall	f1-score	support
1	0.23	0.47	0.31	34
2	0.35	0.21	0.26	34
3	0.19	0.15	0.17	34
4	0.04	0.03	0.03	34
5	0.48	0.85	0.62	34
6	0.63	0.76	0.69	34
7	0.00	0.00	0.00	34
8	0.27	0.35	0.30	34
9	0.61	0.41	0.49	34
10	0.00	0.00	0.00	34
avg / total	0.28	0.32	0.29	340

Accuracy: 0.323529411765

Results using kMeans(80%):

1. For 4 genres:
   a. Accuracy: 75.4%
   b. F1-score: 73%
2. For 10 genres:
   a. Accuracy: 32.3%
   b. F1-score: 29%

Results

Comparison of results for different techniques:

- kNN proved to have the best accuracy at about 91%
- kMeans, as expected, had a lower accuracy of about 75%
- SVM had an accuracy of 90%
- Results seem consistent with previous work done in this field
- Accuracy decreased rapidly with increase in number of categories
- Using CNN: 58% accuracy for 10 genres

What can we do next?

We can further extend our project to map images to songs, or genres in general. CNN can give good results for this work.