Acoustic Scene Classification
Challenge SD 207
Wei Chen, Luo Xi

1. Feature Extraction

The first step in any acoustic recognition system is to extract features i.e. identify the components of the audio signal which are appropriate to identify the characteristics and discarding all the other unimportant information like background noise.

1.1 MFCC

Mel Frequency Cepstral Coefficients (MFCCs) are a feature widely used in acoustic signal recognition, such as_audio scene classification, audio similarity measures etc[1]. It is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency.

With the library ***librosa*** in python, we can easily compute the MFCCs with the function ***librosa.feature.mfcc()***, which will return the MFCCs organised by a series of time frames. The following figure illustrates the characteristic of MFCCs of a signal frame of length X…

1.2 Delta MFCC and delta-delta MFCC

The MFCC feature vector describes only the power spectral envelope of a single frame, but it seems like the audio would also have information in the dynamics i.e. what are the trajectories of the MFCC coefficients over time. So in our approach, apart from 40 MFCCs, we also get 40 delta coefficients and 40 delta-delta coefficients of MFCC, which would combine to give a feature vector of length 120. This can be realized by librosa library function ***librosa.feature.delta( ).***

1.3 CQT

The Constant-Q-Transform (CQT) is a time-frequency representation where the frequency bins are geometrically spaced and the so called Q-factors (ratios of the center frequencies to bandwidths) of all bins are equal [2]. The CQT typically captures 84 bands covering 7 octaves of 12 semi-tones each, however, it allows to set a different number of bands and also a higher number of bands per octave. In our approach, we use the function also provide in librosa, and adopt the parameters suggested in [3]: **24 frequency bands per octave from 5 to 22050 Hz, and 264 frequency bands in total, thus resulting in 264 features for each sample**.

1.4 Feature fusion

The CQT is essentially a wavelet transform, which means that the frequency resolution is better only for low frequencies. So we try to combine the MFCCs features and the CQT features for a more powerful fusion feature. Thus the fusion feature contains **40 MFCCs, 40 delta MFCCs, 40 delta-delta MFCCs, and 264 CQT coefficients.**

1.5 Parameter tuning

We will find in later classification that the classifiers are very sensitive to the length of FFT when calculating MFCCs, and the hop length which is the number of samples between successive frames. In our work, we intend to implement the FFT without overlapping, so we fix hop_length= n_fft. We will apply the k-fold cross validation to tune these two parameters. The mean CV scores with different n_ftt and hop_length are presented in the following table.

n_ftt =hop_length	512	1024	4096	8192	16384
validation score	0.4332	0.4767	0.5165	0.5328	0.5095

2. Classifier

3.1 MLP

This is the classifier we finally adopted eventually. As the MLP classifier is very sensitive to the ***hidden_layer_sizes***, we use the formula below to determine this parameter.

$$hidden\_layer\_size = n\_samples/n\_features + n\_classes$$

Apply the best estimator obtained in the previous step, fitting with the training samples, and predicting the labels of the validation set. We get the accuracy per frame as follow

```
Validation frame accuracy: 0.580757769264
Training time: 30.485364198684692 seconds
```

3.3 Bagging

Using bagging we will get a more stable result which means less bias and smaller variance.

```
Validation frame accuracy: 0.592337164751
Training time: 212.41785883903503 seconds
```

4. Final approach

4.1 Weighted majority vote

As our system analyses and predicts multiple audio segments per input audio file, the way to predict the label for each file is needed. With majority vote, the predictions are made for each frame processed from the audio file. Then a majority vote is taken on all predicted classes from all frames of the same input file. Different from the traditional majority vote, we consider the max predict probability of each frame as the weight of this vote, and the weighted majority vote determines the label of a file.

With the weighted majority vote we can finally predict the label for each file based on the prediction of each frame. For the validation set we get the accuracy for each class and the confusion matrix in the table below.

Table. Confusion matrix for validation set

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	accuracy
0	17					1	1	1			1					0.81
1		14													6	0.70
2			3				3			13						0.16
3				19												1.00
4					15								4			0.79
5						14				4						0.78
6							19			2						0.90
7								6	9	1	2					0.33
8									18							1.00
9					2					13	1	2				0.72
10								2			21					0.91
11					4				4	1		5	4			0.28
12						3			1	2		9	6			0.29
13				1					1					16	1	0.84
14				5									1		12	0.67

```
Validation frame accuracy: 0.592337164751
Validation file accuracy: 0.6758620689655173
```

We can easily notice that our classifier is quite confused about these scene: **café/restaurant, home, park, residential area.**

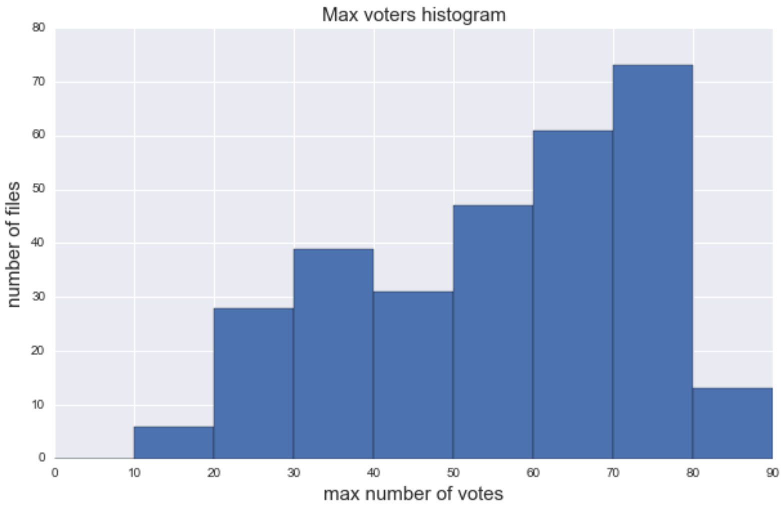
4.2 Expand training data volume

Before applying our classifier to the test data, we try to expand our training data volume by adding the validation data to the training model, which will hopefully boost the performance on test set.

```
Test file accuracy: 0.892617449664
```

4.3 Max voters histogram

To evaluate the prediction on test set, we try to count the number of 'candidates' that vote d for the winner in each file. The more candidates voted for a single class, the more convin cing the result will be. For all the 298 files in test set, the histogram of number of 'votes' is presented. If we assume that more than 40% frames vote for the same class, the predictio n is convincing, then this rate will be 0.8590604026845637, which is quite close to the true accuracy **0.912751677852.**



6 Conclusion

Fusion features containing **120 MFCCs and 264 CQT features** have been applied to a **MLP classifier** for the acoustic scene classification. The optimal accuracy we obtain is **91.28% which ranks 3rd** in the final list.

7 Reference

[1] Meinard Müller (2007). Information Retrieval for Music and Motion. Springer. p. 65. ISBN 978-3-540-74047-6.

[2] C. Sch¨orkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," 7th Sound and Music Computing Conference, pp. 3–64, Jan 2010.

[3] Victor Bisot, Romain Serizel, Slim Essid and Gael Richard, Supervised non negative matrix factorization for acoustic scene classification. Detection and Classification of Acoustic Scenes and Events 2016

[4] CQT-based convolutional neural networks for audio scene classification and domestic audio tagging. Detection and Classification of Acoustic Scenes and Events 2016