Classification and visualization of musical sounds

Mathieu Lagrange

October 17, 2018

This document, the Matlab scripts and the datasets are available here:
https://github.com/mathieulagrange/musnu/tpVisualization

By convention, the Matlab functions that are available as built-in or in the rastamat directory are in bold font.

The "musicGenre" dataset is composed of 10 pairs of music clips. Each pair belong to a given genre: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, Rock.

1 Perceptual mapping

1. Download and decompress the "musicGenre" dataset
2. Use the **napping** function to explore this dataset (clicking nearby a dot allows you to listen to the sound)
3. organize the sounds on the 2D plane in order to have similar sounds that are close to each other and dissimilar sounds that distant.
4. the locations that you set are stored in Comma Separated Value (CSV) file. Copy it to a file with a name containing the name of the dataset and your name.

2 Description

We aim at describing each sound with a set of descriptors, called features. Each features shall be computed on successive overlapping blocks.

1. use a block size of 1024 samples.
2. implement the spectral centroid feature using the **spectrogram** function
3. implement the spectral flatness feature using the **spectrogram** function
4. compute the Mel Frequency Cepstral Coefficients (MFCC)s. To this end, use the **rastamat** or the **librosa** libraries, respectively for matlab and python.

The description for each sound is the averaging over time of the computed features. This means that for the spectral centroid and the spectral flatness, the result is a scalar vector for the whole dataset, one scalar per feature and per sound. For the mfccs, it should result in a 13 times 20 matrix, one vector of 13 scalar values per sound.

3 Visualization

1. Consider the spectral centroid and the spectral flatness as the 2 dimensions for representing the data in 2D.
2. Use the **napping** function to display this projection.
3. Compare qualitatively the organization of the sounds in this "acoustical" plane with respect to the "perceptual" one.

4 Projection

1. Perform a Principal Component Analysis (PCA) of the averaged MFCC features
2. Get the coordinate of the sounds into a 2D place that maximizes the displayed feature variance
3. Use the **napping** function to display this projection.
4. Compare qualitatively the organization of the sounds in this new "acoustical" plane with respect to the "perceptual" one.

5 Extra: classification

We aim at classifying the sounds into their corresponding classes. To do so, we consider a 1 Nearest Neighbor (1-NN) approach using the Euclidean distance. Perform the pooling operation over several texture windows. Prediction is done by majority voting over the several texture window. An example, suppose that your sound is of 12 frames and your texture window is of size 4, the 12 frames are averaged 4 by 4 to give 3 features. Those 3 features a compared to the computed features of the other sounds of the database. Suppose the predicted classes are Rock, Pop, Rock, a majority vote would choose for the Rock class.

1. For each feature, compute the prediction accuracy, that is the number of sound for which the closest sound is of the same class.
2. Does a feature normalization improves the results ?
3. For the MFCCs, shall the 1st coefficient be kept ?
4. Which features gives the best accuracy ?
5. Is it beneficial to combine features ?

6 Report

Please write a report using your favorite word processing tool and output a pdf file. The report shall have for each question a brief description about the way things have been done and some discussion about the resulting behavior.

Send an archive containing the report, the code and the csv files no later than an hour after the end of the session to `mathieu.lagrange@cnrs.fr`, with the `[ECN]` flag within the title of the message.

A Useful commands

A.1 Miscellaneous

- hist : histogram
- repmat : matrix replication
- imagesc : scaled matrix display

A.2 Distance

- pdist : pair wise distance computation
- squareform : convert output of pdist from vector to matrix

A.3 Documentation

- doc "command" : documentation of "command"
- lookfor "keyword" : show commands with "keyword" in the description