MIREX 2015: METHODS FOR SPEECH / MUSIC DETECTION AND CLASSIFICATION

Nikolaos Tsipas Lazaros Vrysis Charalampos Dimoulas George Papanikolaou
Laboratory of Electroacoustics and TV Systems
Aristotle University of Thessaloniki, Greece
nitsipas@auth.gr lvrysis@auth.gr babis@eng.auth.gr pap@eng.auth.gr

ABSTRACT

With this submission, a set of ensemble learning based methods for the MIREX 2015 Speech / Music Classification and Detection task is proposed and evaluated. The main algorithm for the Detection task employs a self - similarity matrix analysis technique to detect homogeneous segments of audio that can be subsequently classified as music or speech by a Random Forest classifier. In addition to the main algorithm two variations are proposed, the first one employs a silence detection algorithm while the second one omits the self-similarity information and relies solely on the Random Forest classifier. For the Classification task two variants are proposed, both based on a sliding-window classification approach. In the first case a pre-trained model is used, while in the second case, a training phase exploiting training data provided during the submission evaluation, precedes classification.

1. INTRODUCTION

The Speech/Music Classification and Detection task introduced for the first time in MIREX 2015 is organised as two distinct subtasks. The classification task is defined as the binary problem of classifying pre-segmented audio data to the speech or music class. Each evaluated segment is 30 seconds long and contains either speech or music data, mixed (speech over music) segments are not allowed. The detection task is focusing on finding segments of music and speech in a signal (i.e. finding segment boundaries) and classifying each segment as music or speech. The detection algorithm is evaluated on recordings from archives, which are typically at least several minutes long and contain multiple segments.

The rest of the paper is organized as follows: in Section 2 the audio feature extraction and pre-processing procedures are described; in Section 3 a detailed description for the detection task work-flow is presented, while in Section 4 the algorithm used for the classification task is analysed. In Sections 5 and 6 information about the submission packaging is provided and relevant references are included.

2. AUDIO FEATURES AND PREPROCESSING

The features presented in Table 1 are extracted from the audio signal using a Hanning sliding window with step and block size equal to 1024 samples at 44100 Hz sampling rate. Along with a classical audio feature extraction strategy, a temporal feature integration methodology was implemented and evaluated using audio feature statistics of aggregated windows. In particular, the extracted features are aggregated in groups of 64 frames and the mean and standard deviation values are calculated. The resulting feature vectors have a time resolution of 1.48 seconds (block and step size equal to 65536) and consist of 74 components. Feature selection was based on their successful integration in similar speech/music discrimination tasks [6] [4] [5] [2] and event detection algorithms [8]. Furthermore the performance of the selected feature set was evaluated through a set of experiments using a trial and error process.

Feature	**Dimension**
RMS Energy	1
ZCR	1
SpectralRolloff	1
SpectralFlux	1
SpectralFlatness	1
SpectralFlatnessPerBand	19
MFCC	13
Sum	37
Aggregated (*mean, std*)	74
Aggregated + PCA	8

Table 1. Extracted Features

As a preprocessing step, the extracted feature vectors are scaled in order to have zero mean and standard deviation equal to one for each component. Finally, a linear kernel Principal Component Analysis algorithm, aiming to improve generalisation and decrease processing requirements, is applied to reduce the dimension of the final feature vectors to 8 components.

3. DETECTION TASK WORKFLOW

The main processing steps of the speech/music segment detection algorithm are illustrated in Figure 1 and discussed in the following section.

3.1 Machine Learning Model

A Random Forest binary classifier configured with 10 estimators (trees) is used for frame level classification; a choice that was made after evaluating multiple classifier configurations using a grid search approach. The binary classification model was generated using training data from the Mirex 2015 muspeak sampe dataset, the GTZAN speech / music dataset and manually annotated youtube videos. The total duration of the available annotated training data is captured in Table 2.

Speech Data	Music Data
3h 40m	2h 19m

Table 2. Annotated training data duration

K-Fold cross validation was used to evaluate the performance of the model. The average f1-score of the model during a 5-fold cross validation was 0.97.

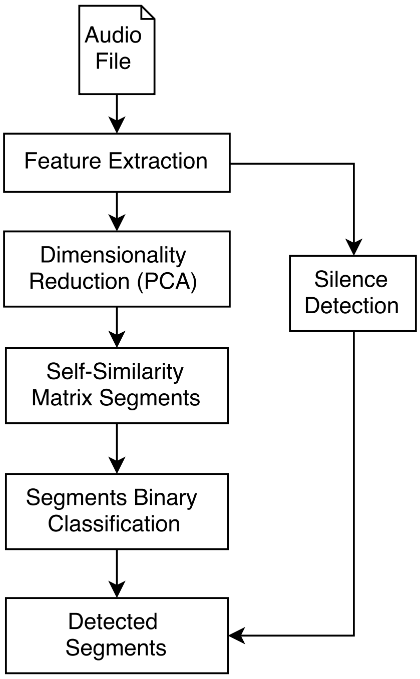
Audio File
Feature Extraction
Dimensionality Reduction (PCA)
Silence Detection
Self-Similarity Matrix Segments
Segments Binary Classification
Detected Segments

Figure 1. The main processing steps of speech/music detection algorithm (TVDP1)

3.2 Self-Similarity Matrix Analysis

The trained model described in the previous section is an integral component of the detection pipeline, however its time resolution is limited to the length of the selected windowing parameters. In order to fine tune the temporal accuracy of the segmentation a self-similarity matrix analysis step was introduced in the processing pipeline. Self-similarity matrices find application in many time series analysis tasks by exploiting similarity information between data samples. In the area of music information retrieval, the analysis of self-similarity matrices is an established approach in song structure segmentation tasks [7]. A self-similarity matrix can be generated by calculating the distance between every pair of samples in the audio signal. In this case the Euclidean distance metric was used to calculate the distance between the extracted feature vectors. An example generated self-similarity matrix is illustrated in Figure 2. Lower intensities (dark pixels) indicate higher pairwise similarity and similarly dark rectangular areas indicate homogeneous segments of audio. By correlating a checkerboard filter [1] with the main diagonal of the generated self-similarity matrix it is possible to detect the transition points between the homogeneous areas. Afterwards, a peak detection algorithm can be used to detect the boundaries of the homogeneous segments as illustrated in 2.

3.3 Segment Classification and Filtering

The frames included in the detected segments are classified using the trained ensemble model. Each frame is classified and the modal value of the classifications is the class assigned to the analysed segment. The above classification step serves also as a filter as it eliminates frames that were misclassified by favoring the most popular class in the segment. Finally, another filtering stage is applied in which adjacent segments of the same class are merged to form a bigger segment.

3.4 Silence Detection

A silence detection algorithm is used to improve the accuracy of the speech / music segments detection algorithm. A fixed threshold t is applied on the RMS Energy of the signal which is extracted with step and block size equal to 1024 samples (at 44.1 kHz). The detected silence segments are added as a final step on top of the detected speech / music segments. The value of the threshold t was set by calculating the mean RMS energy over manually annotated silence audio content.

3.5 Detection Algorithm Versions

- **TVDP1**: Detection with Self-Similarity matrix and Silence detection
- **TVDP2**: Detection with Self-Similarity matrix
- **TVDP3**: Detection without Self-Similarity matrix

4. CLASSIFICATION TASK WORKFLOW

The algorithm for the classification task is based on components developed as part of the detection algorithm. There are two versions of the algorithm, the first one is using an pre-trained model for classification while the second one requires a training phase, exploiting training data provided during the submission evaluation, and precedes classification.

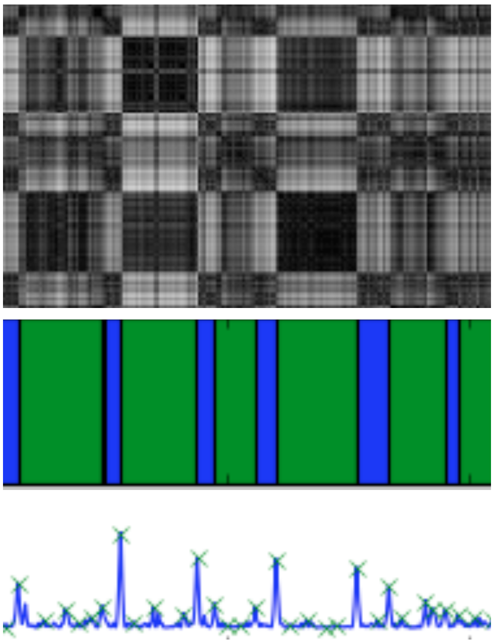

Figure 2. Self similarity matrix analysis example. Ground truth data displayed in the middle. Detected peaks after 2D correlation with a checkerboard filter displayed at the bottom.

4.1 Machine Learning Model

The same Random Forest binary classifier described in the detection task is used for frame level classification. In the case of the pre-trained model variant the same dataset presented in Table 2 is used for training. In the second case, where a training phase is executed as part of the evaluation, Sac [1] is employed to read the extracted audio features and generate the training dataset on the fly.

4.2 Audio File Classification Method

The first processing step for each audio file given as input to the classification algorithm is the extraction of the audio features present in Table 1 through the procedure described in section 2. Afterwards each frame is classified as speech or music and the final class, for the whole file, is derived by majority vote from the classification results of all blocks.

4.3 Classification Algorithm Versions

- **NT1**: Classification with Training
- **NT2**: Classification without Training

5. SUBMISSION PACKAGING

In order to simplify the submission procedure for us and the competition organizers the Docker [3] containerization technology is utilized. As illustrated in Figure 3 the proposed speech / music detection and classification algorithm along with all its dependencies is delivered as an Ubuntu linux system embedded into a docker container. The only required input during the evaluation of the algorithm is the evaluation data locations on the host machine. This submission is available as an open source project on github [2] .

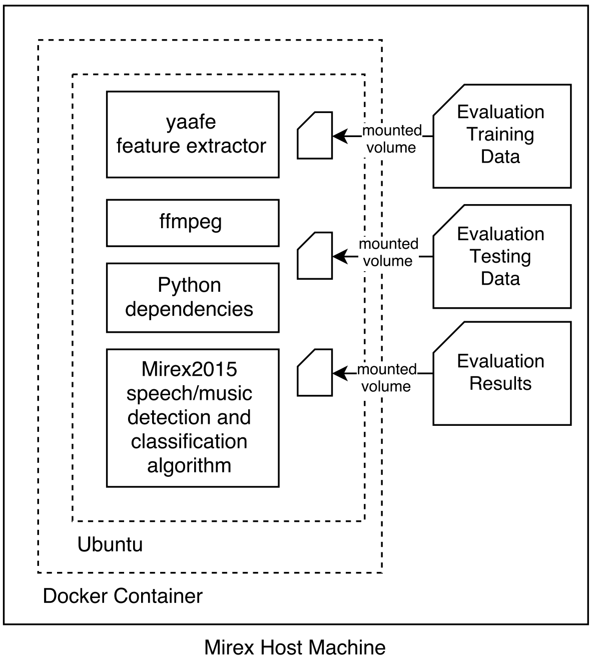

Figure 3. Submission Packaging using Docker

6. REFERENCES

[1] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.

[2] Rigas Kotsakis, George Kalliris, and Charalampos Dimoulas. Investigation of broadcast-audio semantic analysis scenarios employing radio-programme-adaptive pattern classification. *Speech Communication*, 54(6):743–762, 2012.

[3] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239):2, 2014.

[4] John Saunders. Real-time discrimination of broadcast speech/music. In *icassp*, pages 993–996. IEEE, 1996.

[5] Eric Scheirer and Malcoh Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1331–1334. IEEE, 1997.

[6] Nikolaos Tsipas, Zapartas Panagiotis, Lazaros Vrysis, and Charalampos Dimoulas. Augmenting social multimedia semantic interaction through audio-enhanced web-tv services. In *Audio Mostly*, 2015.

[1] https://github.com/nicktgr15/sac

[2] https://github.com/nicktgr15/mirex2015

[7] Nikolaos Tsipas, Lazaros Vrysis, Charalampos Dimoulas, and George Papanikolaou. Content-based music structure analysis using vector quantization. In *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[8] Lazaros Vrysis, Nikolaos Tsipas, Charalampos Dimoulas, and George Papanikolaou. Mobile audio intelligence: From real time segmentation to crowd sourced semantics. In *Audio Mostly*, 2015.