AUTOMATIC EVENT CLASSIFICATION USING FRONT END SINGLE CHANNEL NOISE REDUCTION, MFCC FEATURES AND A SUPPORT VECTOR MACHINE CLASSIFIER

Waldo Nogueira, Gerard Roma and Perfecto Herrera

Music Technology Group
Universitat Pompeu Fabra
Roc de Boronat 138,
08018, Barcelona
waldo.nogueira@upf.edu

ABSTRACT

This submission to the sub-task scene event classification Office Live of the IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events uses first a single channel noise reduction to clean stationary background noise, next mfccs are extracted and finally a support-vector machine classifier is used to classify the events. In this short paper the usage of the implementation as well as a short description of the system is explained.

Index Terms— mfcc, support vector machine, event detection, noise reduction

1. INTRODUCTION

The system designed uses a feature selection unit based on mfccs to train a support-vector-machine (SVM) classifier 1. In parallel to this classifier we use an identical system that uses front end preprocessing module based on single channel noise reduction algorithm. Figure 2 present the basic block diagrams of this system. Both systems have been combined to improve the overall recognition rate.

The predicted labels of both systems are finally combined into one single set of predicted labels.

2. METHODS

2.1. Front End

A front-end single channel noise reduction has been implemented in order to reduce the negative effects of background noise in the event classification performance. The single channel noise reduction algorithm uses minimum statistics to estimate the background noise in different frequency bands. Next an estimation of the signal-to-noise ratio is performed. The target signal is whatever is not stationary in each frequency band and noise is whatever is stationary in each frequency band. Finally a Wiener filter is applied to each frequency band in order to maximize the signal-to-noise ratio.

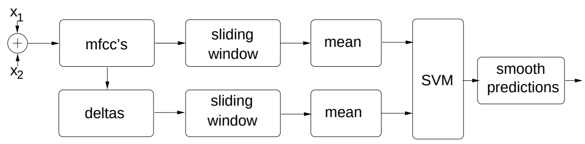

Figure 1: Block diagram for the event detection algorithm.

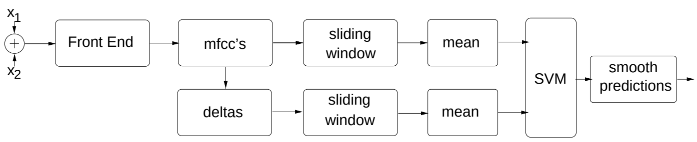

Figure 2: Block diagram for the event detection algorithm with Front End Processing.

2.2. Mel Frequency Cepstrums

The spectral features are based on the commonly used mfcc. The stereo soundtracks formed by the left and right channels (x_1 and x_2) sampled at 44.1 kHz are mixed by taking the average of each sample between the left and right channel and processed using a short-time Fourier magnitude spectrum calculated over a 20-ms window every 10 ms. The spectrum of each window is converted into to the Mel frequency scale, and the log of the Mel bands is computed. Finally, discrete cosine transform (DCT) is applied to the Mel bands to decorrelate the data. For each audio file we took 13 mfccs.

In order to reduce the amount of frames a sliding window of 10 frames with 5 frames overlap was used. For each window the mean and mean of the deltas of the mfcc were extracted and used as features to train the SVM.

2.3. Combination of the systems

The event classifier with noise reduction front end seems to be able to classify much better what belongs to the background. However, the system without Front-End processing seems to be better at identifying the different events, although it creates many false alarms when there is no event present. From these results we decided to combine the predicted outputs from both systems such that the system with front end processing is used to decide whether the current frame is background or not. On the other hand, the system without front end processing is used to decide which class is whenever the

Table 1: Results for the frame based metrics

Rec	56.21 %
Pre	22.85 %
F	32.49 %
AEER	2.7740

Table 2: Results for the event based metrics

Rec	31.43 %
Pre	6.71 %
F	11.06 %
AEER	5.7429
RecOff	20.00
PreOff	4.27
F0Off	7.04
AEER	6.0857

frame is not classified as background by the system with front end processing.

2.4. Support Vector Machines (SVMs)

For classification we used a state-of-the-art supervised learning method based on SVM [1]. We used a linear distance between the examples to create the gram matrix $K(f, g)$

$$K(f, g) = e^{-\gamma D(f,g)}. \quad (1)$$

We use the so-called slack SVM that allows a trade-off between imperfect separation of training examples and smoothness of the classification boundary, controlled by a constant C that we vary in the range $10^1, 10^2, ..., 10^{10}$. Both tunable parameters γ and C were chosen to maximize the classification accuracy over a set of validation data.

3. EVALUATION

We evaluated our approaches using the database of isolated events for training the algorithm. Part of these database was also used to perform the grid-search of γ and C before performing the classification with SVM.

Next table presents the results obtained using the metrics delivered by the organizers.

4. CONCLUSIONS

In this short paper we have described a method to classify events in the presence of background noise. The solution adopted by the authors is based on extracting the mfccs from a noise reduction front end that cleans the noise incoming sound. The mean of the mfccs is used to train a support vector machine classifier.

Table 3: Results for the class based metrics

Rec	31.81 %
Pre	27.44 %
F	20.90 %
AEER	4.323
RecOff	21.81
PreOff	8.05
F0Off	10.53
AEER	4.62

5. REFERENCES

[1] K. Lee and D. P. Ellis, and E. Roberts, "Audio-Based Semantic Concept Classification for Consumer Video," *IEEE Trans. Audio, Speech and Language Process.*, vol. 18, no. 6, pp. 1406–1416, Aug. 2010.

[2] Courtenay V. Cotton and Daniel P. W. Ellis, "Spectral vs. Spectro-Temporal Features for Acoustic Event Detection" *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.*, October 2011.

[3] `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/mfccs.html`.

[4] Chang, Chih-Chung and Lin, Chih-Jen, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, issue. 3, 2011, pp. 27:1–27:27, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`