# EVENT DETECTION AND CLASSIFICATION

*Sameer Chauhan, Sharang Phadke, Christian Sherland*

Cooper Union for the Advancement of Science and Art
Electrical Engineering
41 Cooper Square
New York, NY 10003

## ABSTRACT

The IEEE AASP Challenge addresses the problem of acoustic event detection and classification in an office environment. Our system performs segmentation and event classification on a continuous stream of acoustic activity in an office using basic feature extraction techniques and a single layer frame-by-frame classifier. We achieve high classification accuracy in noiseless environments, but performance severely deteriorates in noisy environments.

***Index Terms***— onset, offset, MFCCs, LRT

## 1. INTRODUCTION

The task of event detection and classification is fundamentally important in computational auditory scene analysis (CASA). Detecting events such as words in a sentence of speech, the arrival of a bus or train, or any other common occurrence in an audio stream is the first stage in many speech and audio processing systems. Many such systems then need to classify the detected events in order to take appropriate actions.

In this project submission, the problem of event detection and classification, as applied to an office environment, was approached from a pattern recognition perspective. Our system consists of a segmentation stage, a feature extraction stage, and two classification stages. The first stage detects the onset and offset of events in a live recording, the second extracts features from each event that can be used to classify events, and the final stages classify each detected event on a frame-by-frame basis using a pre-trained likelihood ratio test (LRT) classifier with a MAP decision rule. The performance of the system is evaluated on three datasets: a 5-fold cross-validation using the training dataset itself, the development set with "perfect" segmentation, and the raw development set.

## 2. FEATURE EXTRACTION

In order to classify events, our system extracts an extensive set of features from the provided auditory data. This set includes frequency-based features as well as time-based features, which, together, constituted a 39 element feature vector. Each of the features were computed in 50ms windows to balance frequency and time resolution.

### 2.1. Frequency-Based Features

The set of frequency-based features includes the spectral centroid, spectral flux, spectral entropy, short time energy (STE), spectral roll-off, and Mel-frequency cepstral coefficients (MFCCs). Each of these feature highlights different spectral properties of the signal. Short time energy is analogous to the volume of the event and the spectral centroid is essentially the center of mass of the spectrum. They are both expected to be reliable indicators of silence. The spectral flux, also called spectral variation, measures how quickly the power spectrum changes. It can be used to determine the timbre of the audio signal. Finally, the MFCCs, some of the most widely used acoustic features, measure the power spectrum of the signal in one of 20 computed bands equally spaced on the Mel frequency scale. Each of these features were computed with either hamming or rectangular windows. In addition, the mean and standard deviation of each frequency-based feature, excluding MFCCs, was taken across entire training events.

### 2.2. Time-Based Features

Time-based features consisted of loudness, wavelet decomposition coefficients, and autocorrelation. Loudness was measured simply as the rms value of a windowed signal, and provided incremental but significant new information to the system. A three layer wavelet decomposition was implemented using the Discrete Meyer wavelet, and the mean and standard deviation of the wavelet coefficients were computed. The autocorrelation proved to provide the maximum information of any of the time-based features, improving performance by over 15%. The mean, median, and standard deviation of each windowed autocorrelation was taken.

## 3. TRAINING

The classification stage of our system employs a generalized likelihood ratio test classifier. The training data was broken up into frames and a full set of features was extracted from each frame. Each set of features was labeled with its class number and used as training data for the classifier.

## 4. SEGMENTATION

Before events within a continuous input signal can be classified, the segments of the input which contain events must be extracted.

Thanks to the Cooper Union

A speech segmenter [1] accomplishes this by examining the short time energy and spectral centroid features. The segmenter examines peaks of the features and checks them against a threshold to determine the onset and offset of events. The onset and offset times are then used to extract the portions of the input signal which contain an event and pass them on to the classification stage.

A continuous sound file is input into the segmenter which applies a Chebyshev Type II low pass filter. This step was added to the original segmenter to reduce the amount of background noise and increase onset and offset detection. The segmentation process requires three parameters for feature extraction: window length, step time, and the weighting facter that will be used to compare the signal to a threshold. The segmenter was exhaustively tested against the development data to determine the parameters that provided optimal segmentation. Each run of the segmentation test was checked for the number of events it detected. The runs were narrowed down to those that returned the same number of events as the development data indicated. Then the mean square error was calcuated using onset and offset times and used to determine the optimal choice of parameters.The following is the cost function used to measure the segmenter, where $\tau_{(on)i}$ is the estimated onset time and $tau_{(off)i}$ is the estimated offset time and $t_{on}$ and $t_{off}$ are the truth values.

$$MSE = \min \left(\tau_{(on)i} - t_{on}\right)^2 + \left(\tau_{(off)i} - t_{off}\right)^2 \qquad (1)$$

The values for the window length, step time, and threshold weight that results in the minimum MSE were 0.05s, 0.04s and 3 respectively. During the exhaustive process, it was also determined that the threshold weight had to be an integer value.

The segmenting function returns a two column vector where the first column indicates each onset, and the second column indicates each event offset. The signal between the onset and offset times is used in the classification stage of the system.

## 5. CLASSIFICATION

After a set of event signals has been segmented, the classifier extracts a full set of features for each event signal. For each event, these features are passed into the first level classifier. Based upon the output of the first stage classifier, each event is labeled with its subgroup.

Each event signal was then passed through the classifier that corresponds to its subgroup label and reclassified based upon the subset of features that best describes its type of event.

Types were chosen such that members of a given type generally have many similar features. The set of features that is used to determine the specific event that occurred in stage two is the set which varies most among the members.

## 6. RESULTS

In order to evaluate the performance of our system we examined the percentage of frames correctly classified with our segmentation and with perfect segmentation. The perfect segmentation and ground truth label for each frame were determined using the annotation to the development data.

The percentage of frames correct given perfect segmentation gives a sense of how well the classification stage of our system performs. Currently, given perfect segmentation, our system correctly classifies 63% of frames.

The percentage of frames correct with segmentation determined by our system when compared to the other metric gives a good sense of how segmentation affects classification. With our segmentation, the system correctly classified 36% of frames.

The confusion matricies of classifier stages of the perfectly segmented events, as well as the events segmented using our algorithm can be seen in Figure 1 and Figure 2 respectively. (We need to make the figure bigger...)

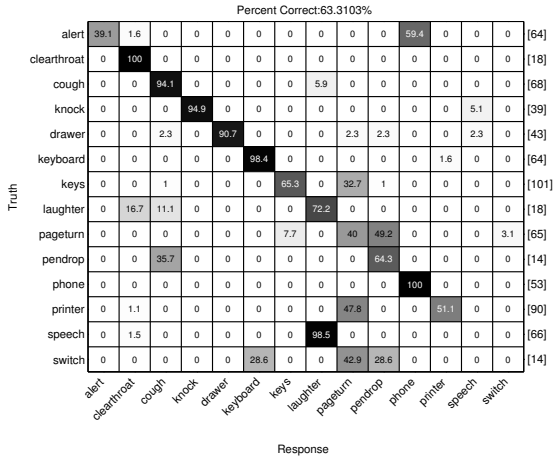


Figure 1: Example of a figure with experimental results.

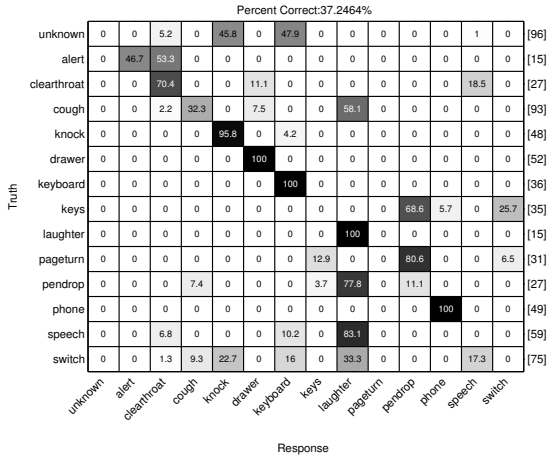


Figure 2: Example of a figure with experimental results.

It is worth noting that the above only hold for frame based classification. Our system was not able to classify events as well as frames. The class of an event was chosen by taking the class chosen most often in the set of frames that formed the event. Because our system is not able to accurately classify more than 50of classifying events does not perform well.

## 7. CONCLUSION

In a k-folds sense, our classification system performed well. This implies that our system is not deficient in features that accurately classify events in this challenge.

The major challenges that hindered the performance of our classification were noise in the development set and the problem of segmenting the input file into individual events. The noise also posed a major challenge in segmenting the file.

## 8. REFERENCES

List and number all bibliographical references at the end of the paper. The references should be numbered in order of appearance in the document. When referring to them in the text, type the corresponding reference number in square brackets as shown at the end of this sentence [**?**], [**?**]. For LaTeX users, the use of the BibTeX style file IEEEtran.bst is recommended, which is included in the LaTeX paper kit available from the workshop website [**?**].

## 9. ACKNOWLEDGMENT

Available Citations: [3] And [1] and [2] and [8] and [7] and [4] and [5] [6] We would like to thank to Professor Sam Keene for his support and guidance. We would also like to thank New Folder Consulting for allowing us to use PRT.

## 10. REFERENCES

[1] Theodoros Giannakopoulos, "A method for silence removal and segmentation of speech signals, implemented in Matlab", *Department of Informatics and Telecommunications University of Athens, Greece—*, 2009. Software available at `https://www.mathworks.com/matlabcentral/fileexchange/28826-silence-removal-in-speech-signals`.

[2] Peter Torrione and Sam Keene and Kenneth Morton, "PRT: The Pattern Recognition Toolbox for MATLAB", 2011. Software available at `http://newfolderconsulting.com/prt`.

[3] Gordern Wichern, Jiachen Xue, Harvey Thornburg, Brandon Mechtley, and Andrewwas Spanias, "Segmentation, Indexing, and Retrieval for Environmental and Natural Sounds" in *"IEEE Trans. Signal Processing*, vol. 18, pp 688–707, March 2010.

[4] Theodoros Giannakopoulo, "Some Basic Audio Features", Apr. 2008. Software available at `http://www.mathworks.com/matlabcentral/fileexchange/19236-some-basic-audio-features`

[5] Pampalk, Elias, "Computational Models of Music Similarity and Their Application in Music Information Retrieval."*Vienna University of Technology*, Mar. 2006

[6] Xiaodan Zhuang and Xi Zhou and Mark A and Hasegawa-Johnson and Thomas S. Huang, "Real-world acoustic event detection", *Pattern Recognition Letters* vol. 31, pp 1543–1551, Feb 2010

[7] Geoffrey Hinton and Li Deng and Dong Yu and George Dahl and Abdel-rahman Mohamed and Navdeep Jaitly and Vincent Vanhoucke and Patrick Nguyen and Tara Sainath and Brian Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," in *IEEE Signal Processing Magazine*, pp. 82–97, Nov. 2012

[8] Xiaodong and Wu Chou. "Discriminative Learning in Sequential Pattern Recognition," in *IEEE Signal Processing Magazine*, pp. 14–36, Sept. 2008.