A Voting-Based Technique for Acoustic Event-Specific Detection

Huy Phan[1,2], Alfred Mertins[2]

[1] Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Email: phan@isip.uni-luebeck.de

[2] Institute for Signal Processing, University of Lübeck, Email: mertins@isip.uni-luebeck.de

Introduction

Acoustic event detection has been an active research topic during last few years. However, building an acoustic event detection system still remains a challenging task. The difficulty stems from the large intra-class variations in terms of different temporal scales and sounds, non-stationary background noise, and, especially, the nature of overlapping events.

Several works attempted to address the problem. In general, these employ simple frame-level presentations and a variety of classification algorithms. Typically, individual events are modelled as Hidden Markov Models (HMM), and a speech recognition framework is employed to detect them [4]. The audio segments can also be characterized by the Gaussian population histograms derived from a Gaussian Mixture Model (GMM), and the detection is performed as classification task using GMMs [5]. In another work, Support Vector Machines (SVM) are directly used over feature vectors derived from audio signals [2].

In this work we introduce a novel concept of *acoustic superframe* and how event detection can be accomplished by recognition of superframes using a simple but efficient class-specific voting scheme. We employ *random forest* [3] to model the event superframes. After detection of individual event superframes, the detection hypotheses for the events will correspond to majority voting from all superframes. The evaluation on the UPC-TALP database from CLEAR 2006 challenge [1] shows that our approach outperforms the best system submitted to that challenge.

The concept of acoustic superframe

Given an audio signal, we divide it into interleaved 100 ms segments with 50% overlap. We call each of these segments a *superframe*. Each superframe is divided into small frames of 20 ms duration with 50% overlap. To represent a frame, we utilized the acoustic feature set suggested by [2] including (1) 16 frequency-filtered log filter-bank energies, along with the first and second time derivatives, and (2) the following set of features: zero-crossing rate, short time energy, 4 sub-band energies, spectral flux calculated for each sub-band, spectral centroid, and spectral bandwidth. Totally, 60 features are extracted for each frame. The mean and the standard deviation of the frame representations are used to represent a superframe, resulting in 120-dimensional feature vector. Each event is considered as a collection of superframes.

This concept can be seen as a mid-level representation which is a trade-off between too noisy frame-wise and too conservative event-wise representations. The superframes are invariant to temporal scale; therefore, we do not have to deal with the scaling issue as event-wise representation. On another hand, they help to reduce the number of data samples generated compared to frame-wise representation and, hence, facilitate the training and testing process in terms of speed.

Random forest for superframe modelling

Since the number of superframes generated from the data is very large, it leads to large-scale problems in both training and testing. For the dataset used in experiments, training and testing data contained 79,586 and 39,312 samples respectively. It is problematic for most popular classification algorithms, such as SVM [6], especially when accompanied with non-linear kernels. Fortunately, random forest [3] is particularly suitable for this purpose since it is efficient for data with large number of samples and dimensions. A random forest classifier consists of a number of trees, each of which is grown using some forms of randomization. The leaves of each tree are labelled by estimates of the posterior distribution of all categories. Each decisive node performs a test to best split the data in feature space. A data sample is classified by sending it down every tree and aggregating the reached leaf distribution.

Let $\{(x_i, y_i)\}_{i=1,\dots,N_{tr}}$ denote the training data where $x_i \in \mathcal{R}^D$ and $y_i \in \{0, \dots, C\}$ denote the feature vector and label of the superframe i, respectively. N_{tr} is the cardinality of training data. $D = 120$ and $C = 12$ are the dimensionality and the number of event categories, respectively. The background is labelled as class 0. Since we want to detect an event by detecting its superframes, for each event category, a one-vs-rest classifier can be built to discriminate the superframes belonging to this category from the rest. However, this strategy is not preferable since the training data would be highly skewed. Alternatively, we build two models: (1) a binary classifier to classify background superframes from foreground superframes; (2) a multi-class classifier to discriminate between superframes of different categories. Since the background is quite different and easy to be distinguished from the events, it is reasonable to deal with it first. Moreover, learning the multi-class classifier can avoid dealing with extremely unbalanced training data.

Event-specific detection system

Let M_{bg} and M_{event} denote the binary background-vs-foreground and multi-class event classifiers that have been trained from training data. The pipeline of our system is shown in Figure 1.

Given a testing signal, we divide it into N_{te} superframes as in Figure 1(a). Each superframe i represented by feature vector $x_i \in \mathcal{R}^D, i \in \{1, \dots, N_{te}\}$ is firstly fed into

Table 1: Performance of event detection task

	Our approach				UPC-D	CMU-D2	ITC-D2
	$W = 5$	$W = 10$	$W = 15$	$W = 20$			
AEER	35.4 %	**32.4 %**	**32.2 %**	**32.1 %**	58.90 %	52.5 %	33.7 %

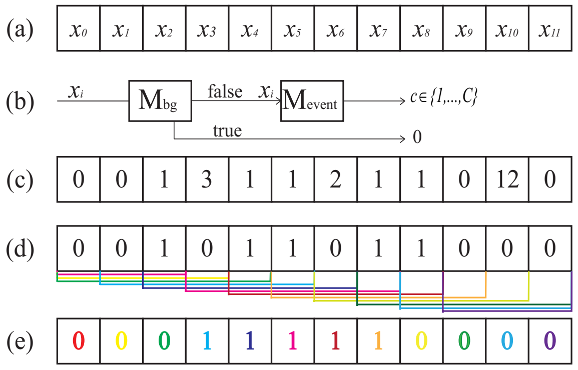

Figure 1: Superframe-based voting event detection system.

M_{bg}. If the superframe is not classified as background, it is subsequently inputted into M_{event} and is finally labelled as class $c \in \{1, \dots, C\}$. The classification process is illustrated in Figure 1(b) whereas Figure 1(c) illustrates the label sequence obtained by classifying the superframes in Figure 1(a). As expected, the sequence appears to be noisy due to the misclassification caused by both M_{bg} and M_{event}. We propose a simple voting scheme to smooth the label sequence. At every superframe i we employ a window of length W centered at i, as demonstrated in Figure 1(d) for the particular case where $c = 1$ and $W = 5$, and update its label by majority voting of all superframes inside the window:

$$\tilde{y}_i = \underset{c \in \{0,\dots,C\}}{\arg\max} \sum_{k=i-\frac{W}{2}}^{i+\frac{W}{2}} \mathcal{I}(\hat{y}_k = c). \tag{1}$$

In (1), $\tilde{y}_i$ denotes the resulting label after majority voting and $\hat{y}_k$ is the predicted label outputted by random forest classifiers. $\mathcal{I}(\hat{y}_k = c)$ is the indicator function given by:

$$\mathcal{I}(\hat{y}_k = c) = \begin{cases} 1 & \text{if} \quad \hat{y}_k = c \\ 0 & \text{if} \quad \hat{y}_k \neq c. \end{cases} \tag{2}$$

The smoothed label sequence is shown in Figure 1(e). Eventually, the subsequences of consecutive event superframes are considered as detection hypotheses.

Experiments

We test our approach on the UPC-TALP dataset of isolated meeting-room acoustic events that were used in the CLEAR 2006 evaluation [1]. It consists of three recording sessions each of which was performed by the same ten actors. The database contains 13 semantic classes: knock (door, table); door open; door close; steps; chair moving; spoon (cup jingle); paper work (listing, wrapping); key jingle; keyboard typing; phone ringing/music; applause; cough; and laughing. About 60 sounds per class were recorded and no overlapping between events was present. As in the CLEAR 2006 evaluation, twelve classes (excluding door close), were evaluated and the rest, including door close, speech, unknown events, and silence were considered as background. We used data from Session 1 and 2 for training and data from Session 3 for testing. Furthermore, only one channel was used in our experiments.

Firstly, the sound signal was down-sampled from 44.1 kHz to 16 kHz and divided into superframes. Each superframe was labelled as $c \in \{1, \dots, 12\}$ if it belonged to the event category c. Otherwise, it was labelled as background. This generated training and testing data with 79,586 and 39,312 samples, respectively. Using random forest [3], we built two models from training data: (1) a binary classifier M_{bg} to classify background samples from foreground samples; (2) a multi-class classifier M_{event} to tell apart event superframes of the 12 semantic classes. For both models, we conservatively set the number of trees to 500.

Following the steps described previously, we performed event detection on the testing data with different windows $W = 5, 10, 15, 20$ which are equivalent to 0.5, 1, 1.5, and 2 seconds respectively. We use the same Acoustic Event Error Rate (AEER) which is given by

$$AEER = (D + I + S)/N, \tag{3}$$

for evaluation, where N is the number of events to detect, D, I, and S correspond to deletion, insertion and substitution errors. The detection results of our approach as well as other systems reported in [1] are tabulated in Table 1. As can be seen, our approach with $W = 10, 15, 20$ outperforms not only the two systems UPC-D and CMU-D2 with a large margin but also the best system ITC-D2.

Conclusion

We introduced in this paper the concept of acoustic superframe to represent an event as a collection of superframes and proposed a simple but efficient voting technique for acoustic event detection based on detection of superframes. The performance evaluation on the UPC-TALP database demonstrates the efficiency of our proposed approach on event detection.

Acknowledgements

This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1].

References

[1] Temko, A., Malkin, R., Zieger, C., Macho, D., Nadeu, C., Omologo, M.: CLEAR evaluation of acoustic event detection and classification systems. Lecture Notes in Computer Science 4122 (2007), 311-322.

[2] Temko, A., Nadeu, C.: Acoustic event detection in meeting-room environments. Pattern Recognition Letters 30 (2009), 1281-1288.

[3] Breiman, L: Random Forest. Machine Learning 45 (2001), 5-32.

[4] Zhuang, X., Zhou, X., Huang, T. S., Hasegawa-Johnson, M: Feature analysis and selection for acoustic event detection. In ICASSP (2008).

[5] Atrey, P. K., Maddage, N. C., Kankanhalli, M. S.: Audio based event detection for multimedia surveillance. In ICASSP (2006).

[6] Smola, A. J., Schölkopf B.: Learning with Kernels. MIT Press (2002).