Consumer-level multimedia event detection through unsupervised audio signal modeling

Byungki Byun1, Ilseo Kim1, Sabato Marco Siniscalchi2, Chin-Hui Lee1

1School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
2Telematics Engineering, Kore University of Enna, Enna, Sicily, Italy
{yorke3, ilseo}@gatech.edu, marco.siniscalchi@unikore.it, chl@ece.gatech.edu

Abstract

In this work1, a novel acoustic characterization approach to multimedia event detection (MED) task for unconstrained and unstructured consumer-level videos through audio signal modeling is proposed. The key idea is to characterize the acoustic space of interest with a set of fundamental acoustic units around which a set of acoustic segment models (ASMs) is built. A vector space modeling technique to address MED is here adopted, where an incoming audio signal is first decoded into a sequence of acoustic segments. Then, a feature vector is generated by using co-occurrence statistics of acoustic units, and the MED final decision is implemented with a vector space language classifier. Experimental evidence on the TRECVID2011 MED demonstrates the viability of the proposed approach. Furthermore, it better accounts for temporal dependencies than previously proposed MFCC bag-of-word approaches. Index Terms: multimedia event detection, unsupervised audio modeling, acoustic segment models

1. Introduction

With the recent proliferation of mobile devices, the amount of multimedia data created by non-professionals is exponentially growing. As a natural response, there has been an increasing interest from research communities to develop a framework that can help users better consume such a large volume of multimedia data. The challenge is that ordinary users tend to search for a particular multimedia example using textual descriptions that discernibly explain the content of the example rather than using low-level signals represented with feature vectors.

Multimedia events are typically defined as complex activities occurring in a specific place and time with a number of human actions, processes, and activities. Interestingly, audio signals often provide crucial information to identify such activities as studied in [1]. One conventional way of exploiting audio information for multimedia event detection is to pre-define a set of audio events that are able to distinguish one event from others [1, 2, 3]. However, using a pre-fixed set of audio key events to perform event detection might not be suitable because consumer-level videos tend to be unconstrained and unstructured and thus, there exists a wider range of variability in audio signals.

Alternative approach to leveraging audio information for multimedia event detection is to build upon the concept that humans acquire audio information in a bottom-up and unsupervised way in a cluttered acoustic scene and subsequently use them to classify the distinctive events in the scene. The main building block of this kind of approach is to construct acoustic units that are designed to capture certain acoustic events through clustering and to index audio signals using the acoustic units. One set of techniques to generate the acoustic units use frame-level feature such as MFCCs, Zero Crossing Rate (ZCR), etc. [4, 5]. A potential problem of the use of such frame-level features is, however, that temporal constraints are very limited in such features for some of the key audio signatures in videos to help to identify an event class.

We, therefore, use acoustic segment models (ASMs) to understand a broader range of semantics of an auditory context by capturing the temporal constraints in audio signals. ASMs are originally proposed to characterize fundamental speech sound units for speech recognition [6]. Then, this idea is extended for various types of audio units to perform music genre classification [7], speaker recognition [8], etc. In this work, we further extend the use of ASMs to build acoustic words for multimedia event detection, which we adopt HMMs to generate the ASMs. Then, audio clips are segmented through Viterbi decoding, and feature vectors for actual detection tasks are subsequently constructed by computing n-gram statistics of the acoustic words as if we are performing a spoken language understanding task. Similar idea was studied in [9] where acoustic words were built by clustering GMM supervectors. However, they neither considered interactions between neighboring acoustic words nor reported detection performances.

Furthermore, even with well-built acoustic units,
identifying multimedia events can be very challenging because we should simultaneously take into account considerably diverse types of sounds (e.g., speech, music, and many other audio footprints) and their combinations. As a result, using linearly represented ASMs in a single acoustic space might not be suitable. To address this issue, we propose to use histogram intersection kernel (HIK) [10] to mitigate the variability of the audio signals in multimedia data.

In all, we show how our framework represents clip-level audio feature through unsupervised audio signal modeling for multimedia event detection tasks. Promising results are shown on a challenging real-world dataset.

2. Proposed framework
In this section, we describe the proposed system in detail. Section 2.1 explains how to create ASMs followed by our method to create feature vectors based on ASM n-grams in Section 2.2. In Section 2.3, we present the use of histogram intersection kernel for event detection.

2.1. Acoustic segment models
Intuitively, ASMs can be considered as a generalized form of phoneme models; phonemes are basic units specifically designed for a linguistic space whereas ASMs are those for more general acoustic spaces. Thus, with ASMs, the first step to construct an auditory context in multimedia data is to transcribe audio events, such as water lapping, laughing, crying, tire squealing, etc. The main issue here is that unlike phonemes where physical units are well-defined, it is difficult to determine which acoustic units we need to build a priori, which makes initialization of HMMs particularly challenging.

We address this issue by constructing crude HMMs initially but then refining the models gradually. Precisely, suppose there are \(N\) audio clips, denoted as \(X = \{x_1, \ldots, x_N\}\). Then, we segment them using a simple energy level-based voice activity detector (VAD) generating \(\bar{X} = \{x_1, \ldots, x_{1K_1}, x_{21}, \ldots, x_{NK_N}\}\), where \(x_{ij}\) represents the \(j^{\text{th}}\) segment in the \(i^{\text{th}}\) audio clip. Here, \(K_i\) corresponds to the number of segments in the \(i^{\text{th}}\) clip. Next, from \(\bar{X}\), we blindly select \(M_0\) segments and train initial ASMs for each segment. Because (a) each segment in \(\bar{X}\) rarely contains a single unique acoustic signature and (b) the lengths of such segments are very diverse, the initial ASMs are confined to be simple so that they have single Gaussian distributions for output probabilities.

To refine the initially built ASMs, we alternatively perform a Viterbi decoding and Baum-Welch estimation until convergence. Specifically, suppose there are \(M_1\) ASMs at time \(t\). Using these ASMs, we find the most probable sequence for \(x_{ij}\) in \(\bar{X}\) and use the labels and boundary information along the sequence to re-estimate the ASMs at time \(t+1\). It should be noted that the length of segments in \(\bar{X}\) are very different from each other as aforementioned, the transition matrix are initialized uniformly when Viterbi search is performed with initial ASMs. This way we can create more homogeneous audio segments. It should also be noted that when HMMs are re-trained, we gradually increase the complexity of the models by adding more mixture components for output probabilities. On the other hand, when the mean vectors of Gaussian mixtures from two different ASMs are close enough, we merge them by relabeling corresponding segments with a single unique label. When this process is converged, the resulting ASMs become our building blocks to vectorize audio clips in multimedia data.

2.2. Vector space modeling
Once ASMs are obtained, each audio clip in a multimedia data is transformed into a feature vector treating each ASM as a basis of a vector space. To this end, we calculate ASM n-grams, obtaining bag-of-sounds vectors similar to bag-of-words (BoW) vectors in information retrieval. More precisely, suppose there are \(M\) ASMs represented as \(t_1, \ldots, t_M\). Then, we first decode audio clips in \(X\) with the ASMs into \(N\)-best Viterbi sequences and then count the number of occurrences of each ASM in the Viterbi sequences (unigram). We further compute the number of co-occurrence of two adjacent ASMs (bigram) because a certain activity in multimedia events is dependent on previous and next activities. This creates a feature vector for \(x_i \in X\), represented as \(c_i = [c_i(t_1), \ldots, c_i(t_M), c_i(t_1, t_1), \ldots, c_i(t_M, t_M)]^T\), where \(c_i(\cdot)\) and \(c_i(\cdot, \cdot)\) represent a unigram count and bigram count for \(x_i\), respectively.

Because multimedia events are quite complex so that the dependency tends to exist between two activities occurred in distance, in this work, we evaluate co-occurrence counts not only for adjacent ASMs but also any pairs of ASMs located within a certain window. To discount ASM pairs appearing in long distance, certain weights are multiplied to the associated bigram counts depending on the distance. Specifically, given two ASMs, say \(t_j\) and \(t_k\), the count \(c_i(t_j, t_k)\) is given by

\[
c_i(t_j, t_k) = \begin{cases} \frac{w^{|l(t_j) - l(t_k)|}}{L} & \text{if } |l(t_j) - l(t_k)| \leq L, \\ 0 & \text{otherwise} \end{cases}
\]

(1)

where \(l(t_j)\) represents the location of the ASMs in a Viterbi sequence and \(L\) indicates maximum distance allowed. For instance, if the decoding result for \(x_i\) is \(t_j, t_3, t_k\) and so on, \(l(t_j) = 1\) and \(l(t_k) = 3\). \(w < 1\) is a positive parameter that determine how fast the weight is decaying. These counts are used instead of regular bigram counts in our vector space modeling.

2.3. Histogram intersection kernel
Given two histograms denoted as \(h = [h_1, \ldots, h_D]\) and \(h' = [h'_1, \ldots, h'_D]\), a HIK \(k(h, h')\) computes

\[
k(h, h') = \sum_{i=1}^{D} \min(h_i, h'_i).
\]

(2)
Given count vectors described in Section 2.2, we evaluate (2), after normalizing the count vectors. The HIK is originally proposed in the computer vision community for object detection [10]. Our preliminary experiments verified the usefulness of HIK on audio signal as well; applying (2) on ASM n-grams significantly outperforms the cases where (a) a linear kernel and (b) an RBF kernel with the cosine similarity are used after tf-idf followed by LSI on ASM n-grams. One of the major benefits of HIK is that histogram intersection provides more robust similarity measure among histograms against distractions by background information [10]. Moreover, we can take into account non-linearity between acoustic footprints and multimedia events using HIK.

Once we calculate (2) for all pairs of audio clips in $\mathcal{X}$, we can then apply any kernelized statistical learning algorithms such as SVMs, Gaussian Process (GP), or kernelized MfMfM for multimedia detection.

3. Experimental results

We evaluate the proposed framework on the TRECVID 2011 MED dataset [11], which is an excellent test-bed for a consumer-level multimedia event detection task because of its large volume (45K video clips) and the unconstrained nature. In the TRECVID 2011 MED task, the goal is to build event detectors for the following 10 classes: Birthday party, Changing a vehicle tire, Flash mob gathering, Getting a vehicle unstuck, Grooming an animal, Making a sandwich, Parade, Parkour, Repairing an appliance, and Working on a sewing project. As in the TRECVID guidelines, we denote them from E06 to E15, respectively. Training data consist of about 150 positive samples for each of the target classes and the additional 5 classes denoted as E01‘E05 (refer to [11] for detail). Moreover there are more than 11K pure negative samples that belong to none of the target events. In the test data of 33K clips, there are 120 positive samples (0.4%) for each class on average, and 31K pure negative samples.

ASMs were configured to be 3-state, left-to-right continuous HMMs with 39-dimensional MFCC vectors. As described in Section 2.1, ASMs were trained with audio segments with variable lengths obtained with the VAD. Specifically, 8 initial segments were chosen from all events (i.e., E01‘E15) and we add 5 segments that were filtered out by the VAD to cover silence, resulting in 125 models. While refining the models, we iterated a Viterbi search and a Baum-welch estimation cycle twice and doubled the number of mixture components until it reached 16. The typical length of decoded segments with ASMs is 100‘200ms. For better coverage of a linguistic space, we included 39 monophone models (3-state, left-to-right HMMs with 16 GMMs) trained on the TIMIT database. In fact, in our framework, one can easily add ASMs trained independently so as to increase the coverage of the acoustic units even more. The parameters, $L$ and $w$, defined in (1), are set to 5 and 0.6, respectively.

Figure 1: $F_1$ score at the operating point. Random performance is very low (average $F_1=0.0070$). Overall, ASMs (average $F_1=0.0767$) outperform MFCC-BoW (average $F_1=0.0691$). Moreover, the fusion (average $F_1=0.0829$) of the two audio features shows consistent improvement.

An MFCC-based BoW (MFCC-BoW) approach similar to [4] was chosen to be our primary baseline system. This system was trained as follows: (a) 39-dimensional MFCCs vectors extracted at every 10ms were quantized into $M$ codewords ($M=2,024$ in our case). (b) uni-gram statistics were calculated. (c) HIK was applied. For ASMs and MFCC-BoW, we used SVMs for classifiers.

Detection operating point was determined according the TRECVID guidelines [11]. In particular, the point where the ratio between the probability of missed detection ($P_{M|D}$) and the probability of false alarms ($P_{F|A}$) was equal to 12.5:1 was chosen.

3.1. Comparison with MFCCs

Figure 1 summarizes the overall experimental results. In Figure 1, the $y$-axis represents $F_1$ score at the operating point. It is clearly seen that our system outperforms the baseline by 11% relatively, compared $F_1=0.0767$ with $F_1=0.0691$. These low values of $F_1$ indicate the difficulty of the task. Note that there is an extreme unbalance between the numbers of positive and negative data (i.e., only 0.4% of test data is positive) and video clips in the data set were recorded in unconstrained environments.

It can be seen that there are notable improvements for E06, E09, E11, E13, E14, and E15, where sound types exhibit more prominent temporal constraints (e.g., singing a birthday song, cheering, laughing, human speech, sounds from motorized tools, and tire spinning). This conforms to our hypothesis that ASM would better model temporal constraints than MFCC-BoW approaches, and more importantly, to perform multimedia event detection, to be able to exploit temporal constraints is desired. The decoding process of ASMs provides an additional benefit that more reliable boundaries of semantic acoustic words can be generated compared to frame-based dense sampling used in MFCC-BoW.

Nevertheless, MFCC-BoW outperforms ASMs in E08 and E10. We have found that there exists even larger variation in audio signatures for those two event classes. For example, Grooming an animal exhibits various types of animal sounds with many different ambient noises. Thus, we believe that modeling these two event classes with a limited number of ASMs (165 in total) might not
be sufficient, which raises an interesting question that what is the appropriate number of ASMs that we need to build to cover sound variation in generic event classes. We have left this issue for the future work.

3.2. Confusing event classes by ASMs

We illustrate the distribution of detected event classes by ASMs at the suggested operating points in Figure 2. In Figure 2, each row represents the normalized number of detected samples for a corresponding event class. Note that most of the probability mass is concentrated on the Null class because of the unbalanceness of the data set. Now, let us examine pair-wise confusing event classes that seem interesting to look at in the context of similarity in audio. One example is E08 and E12, where they contain similar sound types, such as background music, street noise, and crowded people, and these sound types are crucial features to differentiate both event classes from others. E07, E11 and E14 are in another interesting group, where in these event classes, audio signals convey instructional narrations and distinctive noises from hand tools. Based on this analysis, we can pursue hierarchical approaches that we identify event class groups first with coarse detectors and use finer detectors only for event classes within an individual group.

3.3. Fusion between ASMs and MFCC-BoW

We conducted additional experiments that combined the two systems (i.e., ASMs and MFCC-BoW) to see if they are complementary. To this end, a late-fusion approach was used with an MFOm classifier [12]. Figure 1 shows that the fusion of both systems present significant improvement for most event classes. Specifically, the average result ($F_1=0.0829$) is improved by 8.1% from ASMs ($F_1=0.0767$) and 19.9% from MFCC-BoW ($F_1=0.0691$) given the operating point chosen as aforementioned. This confirms that ASMs and MFCC-BoW systems are indeed complementary although both are based on the MFCCs features. It is interesting to see that this fusion result is somewhat comparable to the state-of-the-art video feature based systems as reported in [13].

4. Conclusions

In this work, we have presented our novel acoustic characterization framework for consumer-level multimedia event detection. In particular, we showcased an effective scheme to characterize audio signal with a set of ASMs, where feature vectors are constructed by n-gram statistics of decoded ASMs and HIK is applied for better discrimination. Our experimental results on a challenging large-scale consumer video archive are promising, and suggest that our work provides notable usefulness along with conventional MFCC-BoW.

5. References