Compact Audio Representation for Event Detection in Consumer Media

Xiaodan Zhuang, Stavros Tsakalidis, Shuang Wu
Pradeep Natarajan, Rohit Prasad, Prem Natarajan

Speech, Language and Multimedia Business Unit
Raytheon BBN Technologies
10 Moulton Street, Cambridge, MA, U.S.A.
{xzhuang, stavrosa, swu, pradeepn, rprasad, pnataraj}@bbn.com

1. Introduction

The increasing amount of consumer media calls for compact media indexing and effective media search. Most consumer media contain a stream of audio information. Due to its relative invariance with respect to video quality and viewpoint, and the complementary information not available in video, audio is highly informative for consumer media query and retrieval. While human speech content contains salient information related to the topics of the video, it is not always available in consumer media. Indexing the overall audio information using an effective representation is therefore critical.

It is important for the representation to be general enough to support search for various events or topics, while being compact enough to enable efficient storage and access. One popular method views media data as a set of local descriptors. In particular, the soft quantization histogram [1] first assigns each descriptor into the predefined codewords with a soft weight, and then pools the weights for each codeword to construct a codeword histogram, hence often referred to as the bag-of-words representation. It has been successfully used for both audio and vision in many pattern recognition problems [1, 3]. Having a large number of codewords is important for good performance, but vulnerable to heterogeneous data and overfitting issues.

This work presents the application of a compact audio representation for event detection in consumer media. In contrast to the soft quantization histogram, the local descriptors are fit to a parametric distribution, the Gaussian Mixture Model. Motivated by the recent development of “i-vector” in the speaker verification research [2], we further constrain the GMM parameters to a subspace where the most total variability of the descriptors are preserved. In this subspace, referred to as the “total variability” space [2], the GMM parameters construct a compact vector representation robust to data sparsity.

Using the “i-vector” for audio representation in consumer media search has a few merits. First, the i-vector preserves only the subspace of the local descriptor distribution that has the most variation, achieving compactness and robustness to noise. Second, same as the histogram generation, the i-vector generation does not require annotation and can be estimated in an unsupervised fashion.
By increasing unannotated audio data, the i-vector can learn a subspace that is applicable to a diversity of search tasks.

We propose combining the “i-vector” with cosine kernels or chi-square kernels in binary SVMs for detecting a set of heterogeneous events in the audio track of consumer video.

This method is compared against the commonly used bag-of-words soft quantization histograms based on the same MFCC descriptors, as implemented in the state-of-the-art BBN VISER system [3] in the 2011 TRECVID Multimedia Event Detection (MED) Evaluation [4]. We report audio-only event detection performances on all ten target events in 2011 TRECVID MED, and demonstrate that the presented method outperforms the soft quantization histogram with reduced dimensionality.

2. Audio Representations

We represent the audio signal of a video clip as a set of local descriptors $X = \{x_i| i \in \{1 \cdots N\}\}$. The distribution of these descriptors characterises the clip. As in many audio processing works, the local descriptors used in this work are extracted from temporal sliding windows of the same window size and the same step size. However, all methods presented in this work can be applied to local descriptors of other styles, e.g., from localized areas in the temporal-frequency domain.

The essential measure used in most pattern recognition problems is the similarity between two clips. In particular, such measures characterize the similarity between the two distributions of local descriptors from the two clips respectively.

2.1. Soft Quantization Histograms

One commonly used approach to compare the two distributions parameterizes them as two soft quantization histograms. A finite set of representative codewords $\{c_k| k \in \{1..K\}\}$ are identified through clustering or random sampling. Within each audio clip, the “coding” step assigns each local descriptor to one or multiple codewords, denoted by the operator $f$. The “pooling” step produces a numerical representation for each codeword according to descriptors assigned to that codeword, denoted by the operator $g$. This is often referred to as the bag-of-words representation.

$$
[\alpha_{i,1} \cdots \alpha_{i,K}] = f(x_i) \quad i = 1..N
$$

$$
b_k = g(\{\alpha_{i,k}| i \in N\}) \quad k = 1, \ldots, K
$$

Soft quantization [1], as a coding strategy, distributes a local descriptor $x_i$ to a codeword $c_k$ in the following way,

$$
\alpha_{i,k} = \frac{\exp(-\beta||x_i - c_k||^2)}{\sum_{j=1}^{K} \exp(-\beta||x_i - c_j||^2)}.
$$

where $\beta$ controls the smoothness of the soft assignment.

Further, average pooling takes the average of the $\alpha_{i,k}$’s among all $i$ to obtain the final value for the $k$th bin in the histogram: $h_k = \frac{1}{N} \sum_{i} \alpha_{i,k}$.

2.2. I-vector

While the histogram is an effective non-parametric way to summarize the local descriptor distributions, an effective alternative is to use a parametric statistical model, e.g., the Gaussian Mixture Models (GMM).

The GMM supervector method [5] engages a linear approximation of the Kullback-Leibler distance $D(g_a||g_b)$ between two GMMs $g_a$ and $g_b$. It has a simple form below, when the clip-level GMMs are obtained by adapting the mean vectors of a shared Gaussian mixture Universal Background Model (UBM),

$$
D(g_a||g_b) \approx \frac{1}{2} \sum_{k=1}^{K} w_k (\mu_a^k - \mu_b^k)^T \Sigma_k^{-1} (\mu_a^k - \mu_b^k).
$$

$\mu_a^k$ and $\mu_b^k$ denote the adapted means of the $k$th component from the clip-level GMMs $g_a$ and $g_b$. $w_k$ and $\Sigma_k$ are the mixture weight and the covariance matrix. We can consider the corresponding kernel function as a linear kernel between the normalized GMM supervectors $\phi$ in a high-dimensional feature space,

$$
\phi(a) = [\sqrt{\frac{w_1}{2}} \Sigma_1^{-\frac{1}{2}} \mu_1^a; \cdots; \sqrt{\frac{w_K}{2}} \Sigma_K^{-\frac{1}{2}} \mu_K^a].
$$

In the speaker recognition literature, Joint Factor Analysis (JFA) [6] considers the following generative model in the above high-dimensional GMM supervector space:

$$
\phi = m + V y + U x + D z,
$$

where $m$ is the speaker/session-independent component; $V y$ is the speaker subspace component; $U x$ is the session subspace component; and $D z$ is the residual subspace component. First, the corresponding eigenvector matrices $V, U, D$ are estimated using data with speaker annotations. Then the corresponding factors $y, x, z$ are estimated for each utterance.

However, the training annotations are expensive to obtain and it is non-trivial to estimate $V, U, D$ separately [6]. In the recent speaker verification literature [2], the i-vector is proposed as an alternative, where the GMM supervector is confined into a single “total variability” subspace shown in Equation 6, without distinguishing the different sources of variation (speaker, channel, etc).

$$
\phi = m + Tw
$$

The general mean vector above, $m$, is taken directly from the UBMM. Let the factor $w$ observe a standard normal distribution $N(0; I)$, then $\phi$ observes $N(m; TT^*)$. 

INTERSPEECH 2012 2090
The factor $w$ is referred to as the i-vector, as it has a dimensionality between those of the local descriptors and the GMM supervectors.

In this work, we adopt the i-vector approach as a general unsupervised audio modeling method. We estimate the total variance matrix $T$ directly using unlabeled audio that covers a wide range of different target events as well as background data. The total variance matrix is estimated using the same EM algorithm as the eigenvoice matrix in the speaker recognition literature [7], except that each audio clip carries a unique label in replacement of the speaker label. Details of the EM algorithm can be found in [7], and we highlight the following two steps in each iteration:

First, the factor $w$ is estimated for the current clip, according to its posterior

$$P(w|X, \Sigma, T) \sim N(t^{-1}(X)T^T \Sigma^{-1} S_X(X), t^{-1}(X)),$$

where $\Sigma$ measures the deviation from the clip-dependent mean vectors and

$$l(X) = I + T^T \Sigma^{-1} N(X)T.$$  \hspace{1cm} (8)

$N(X)$ is a block diagonal matrix of the number of frames accounted for by the mixtures $N_1(X)I, ..., N_K(X)I$. $S_X(X)$ is the concatenation of the means on the $K$ mixtures, offset by the corresponding means in the UBM.

Second, the total variance matrix $T$ and the matrix $\Sigma$ are updated using linear regression with $w$ being the explanatory variable.

Given the estimated $T$ and $\Sigma$, estimating the factor $w$ for an audio clip is identical to the first step above. Note that no annotation is needed either in estimating $T$ and $\Sigma$, or in estimating $w$ for each clip.

3. Experiments

3.1. Dataset and setup

We test the approach on a large dataset of approximately 10,000 video clips used in the development for the 2011 TRECVID MED Evaluation [4]. This set contains 100-200 examples for each of the ten target events (E006-E015), and the rest are the background. The detection of all the ten target events are carried out separately, i.e. the detection result for one particular event can not be used to inform the detection of other events. The set is split into the non-overlapping training and testing sets with size ratio of 3:1. Only the audio information is used in the experiments.

For each event, a binary SVM with either the cosine kernel or the chi-square kernel is trained to distinguish between clips with target events and other clips. The parameters of the SVMs are estimated through cross validation within the training set [3].

3.2. Local audio descriptor

MFCC is used as the local audio descriptor for all these experiments. Features are extracted from overlapping frames of audio data, each 29 ms long, at a rate of 100 frames per second. Each frame is windowed with a Hamming window and the power spectrum is computed for the frequency band of 80-6000 Hz. From this, 14 Mel-warped cepstral coefficients are computed. The coefficients in each audio clip are normalized by the mean cepstrum and peak energy non-causally, removing any long term bias due to the channel. In addition, they are scaled and translated for zero mean and unit variance in each clip. These base cepstral coefficients with their first and second order derivatives, together with the energy and its first and second order derivatives, compose the 45-dimensional descriptor.

3.3. Audio representations

The i-vector is of 400 dimensions, using a UBM with 512 Gaussians. Two versions of the soft quantization histograms are evaluated, with 400 and 4000 dimensions respectively, the second being the optimized setup used in the BBN VISER submission to the MED task of the 2011 TRECVID Evaluation [3].

3.4. Results

Table 1 shows the Area Under the Curve (AUC) measures of the ten events, for different audio representations, using the cosine kernel. “i-vector” refers to the method presented in this paper. “MFCC” and “MFCC*” refer to the soft quantization histograms with 400 and 4000 dimensions respectively.

Table 2 shows the performance using the chi-square kernels. Note that we linearly shift and scale values in each i-vector dimension so that the normalized feature values fall into the range of $[0, 1]$ for the chi-square kernels.

From Table 1, we can see that, when using a cosine kernel, the proposed method in this work outperforms the MFCC soft quantization histogram in all the ten target events, not only when they both have 400 dimensions, but also when the histogram has an advantage by growing to 4000 bins, the setting optimized for this task in the 2011 TRECVID Evaluation. Table 2 shows the same trend for most of the ten events, when the chi-square kernel is used.

Figure 1 and Figure 2 illustrate detection performance for events observing the most gains and the most hits when comparing the i-vectors with the histograms.

It’s worth noting that the i-vector of 400 dimensions with the cosine kernel outperforms the MFCC soft quantization histograms of a much higher dimensionality with the chi-square kernel.
### Table 1: AUC of event detection using cosine kernels for MED11, the best performances in each column indicated with bold fonts.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dim</th>
<th>Average AUC</th>
<th>E006</th>
<th>E007</th>
<th>E008</th>
<th>E009</th>
<th>E010</th>
<th>E011</th>
<th>E012</th>
<th>E013</th>
<th>E014</th>
<th>E015</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC*</td>
<td>4000</td>
<td>0.772</td>
<td>0.825</td>
<td>0.668</td>
<td>0.890</td>
<td>0.735</td>
<td>0.597</td>
<td>0.802</td>
<td>0.788</td>
<td>0.732</td>
<td>0.901</td>
<td>0.785</td>
</tr>
<tr>
<td>MFCC</td>
<td>400</td>
<td>0.735</td>
<td>0.814</td>
<td>0.642</td>
<td>0.860</td>
<td>0.725</td>
<td>0.555</td>
<td>0.717</td>
<td>0.770</td>
<td>0.632</td>
<td>0.894</td>
<td>0.744</td>
</tr>
<tr>
<td>i-vector</td>
<td>400</td>
<td>0.808</td>
<td>0.826</td>
<td>0.734</td>
<td>0.918</td>
<td>0.778</td>
<td>0.647</td>
<td>0.751</td>
<td>0.819</td>
<td>0.827</td>
<td>0.933</td>
<td>0.853</td>
</tr>
</tbody>
</table>

### Table 2: AUC of event detection using chi-square kernels for MED11, the best performances in each column indicated with bold fonts.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dim</th>
<th>Average AUC</th>
<th>E006</th>
<th>E007</th>
<th>E008</th>
<th>E009</th>
<th>E010</th>
<th>E011</th>
<th>E012</th>
<th>E013</th>
<th>E014</th>
<th>E015</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC*</td>
<td>4000</td>
<td>0.787</td>
<td>0.843</td>
<td>0.732</td>
<td>0.910</td>
<td>0.764</td>
<td>0.647</td>
<td>0.749</td>
<td>0.771</td>
<td>0.816</td>
<td>0.887</td>
<td>0.752</td>
</tr>
<tr>
<td>MFCC</td>
<td>400</td>
<td>0.770</td>
<td>0.844</td>
<td>0.647</td>
<td>0.895</td>
<td>0.791</td>
<td>0.628</td>
<td>0.744</td>
<td>0.796</td>
<td>0.699</td>
<td>0.879</td>
<td>0.780</td>
</tr>
<tr>
<td>i-vector</td>
<td>400</td>
<td>0.814</td>
<td>0.830</td>
<td>0.734</td>
<td>0.927</td>
<td>0.773</td>
<td>0.636</td>
<td>0.760</td>
<td>0.815</td>
<td>0.826</td>
<td>0.947</td>
<td>0.888</td>
</tr>
</tbody>
</table>

### 4. Conclusion and Discussion

This work proposes using i-vector as a compact audio representation for event detection in consumer media. The audio signal of a video clip, represented as a set of frame-level MFCC descriptors, is summarized as a relatively low-dimensional vector, constraining the descriptor distribution in a “total variability” subspace.

We combine the i-vector with the cosine kernel or the chi-square kernel in a set of SVMs for detecting the ten target events defined in the Multimedia Event Detection task of the 2011 TRECVID Evaluation. We demonstrate that the i-vector outperforms the soft quantization histogram using either kernel, with reduced dimensionality.

### 5. References


