SPARSE REPRESENTATION BASED ON A BAG OF SPECTRAL EXEMPLARS FOR ACOUSTIC EVENT DETECTION

Xugang Lu[1], *Yu Tsao*[2], *Shigeki Matsuda*[1], *Chiori Hori*[1]

1. National Institute of Information and Communications Technology, Japan
2. Research Center for Information Technology Innovation, Academic Sinica, Taiwan

ABSTRACT

Acoustic event detection is an important step for audio content analysis and retrieval. Traditional detection techniques model the acoustic events on frame-based spectral features. Considering the temporal-frequency structures of acoustic events may be distributed in time-scales beyond frames, we propose to represent those structures as a bag of spectral patch exemplars. In order to learn the representative exemplars, k-means clustering based vector quantization (VQ) was applied on the whitened spectral patches which makes the learned exemplars focus on high-order statistical structure. With the learned spectral exemplars, a sparse feature representation is extracted based on the similarity measurement to the learned exemplars. A support vector machine (SVM) classifier was built on the sparse representation for acoustic event detection. Our experimental results showed that the sparse representation based on the patch based exemplars significantly improved the performance compared with traditional frame based representations.

Index Terms— Sparse representation, acoustic event detection, vector quantization, support vector machine.

1. INTRODUCTION

Acoustic event detection is an important step for audio content analysis and retrieval [1, 2, 3]. The purpose for acoustic event detection is to classify the audio stream with their semantic categories, and locate the time periods when they occur. The detection of acoustic event takes two steps, one is feature representation, and the second is classifier model training which is used for classification. In most acoustic event detection systems, as used in automatic speech recognition (ASR), the Mel frequency cepstral coefficient (MFCC) is used as feature representation. The hidden Markov model (HMM) and support vector machine (SVM) are the two most popularly used models for classifiers [4, 5, 6].

Extracting representative features is essential for pattern recognition tasks. In most studies, classifier modeling for acoustic event detection (either HMM or SVM) is trained with frame based feature representations (e.g, 20 ms frame length) [3, 4, 5]. In ASR, the frame based acoustic features can be mapped to some intermediate labels before it is finally mapped to utterances, such as from phones to syllables or words. However, in acoustic event detection, we do not have knowledge of such kind of intermediate labels. Directly mapping the frame based representation to their semantic categories is not suitable for the acoustic event detection task. In the meanwhile, acoustic events have well organized temporal-frequency structures spanned in many continuous frames (as spectral patches). These temporal-frequency structures are representative features for the underlying acoustic event sources which can be regarded as parts or sub-parts exemplars for acoustic events. In this study we try to automatically learn these temporal-frequency structures and form a bag of spectral exemplars to represent acoustic event patterns. Based on the learned spectral exemplars, a sparse feature representation is extracted based on the similarity measurement to the exemplars. The new representation can be regarded as a new type of intermediate representations of acoustic events.

The idea of using spectral patch based representation for acoustic event detection has already been proposed [7]. In their work, a nonnegative matrix factorization (NMF) learning algorithm was applied on spectral-temporal features. However, learning directly on spectral patches may only explore representations dominated by the second-order statistical structure. Our work is different from theirs. Inspired by the work in image processing [9], a simple k-means vector quantization (VQ) is used to learn spectral patch exemplars. In addition, before applying the VQ, a whitening process is applied to remove the second-order statistical structure in the spectral patches. In this case, the learned exemplars are much more representative for patterns with high-order statistical structure than using the second order statistic structure. Based on the exemplars, a sparse representation is constructed which is used for modeling and classification.

2. SPARSE FEATURE EXTRACTION BASED ON A BAG OF EXEMPLARS OF SPECTRAL PATCHES

In this section, we introduce how the bag of exemplars of spectral patches for acoustic events is learned, and how the sparse representation feature is extracted based on the learned exemplars.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

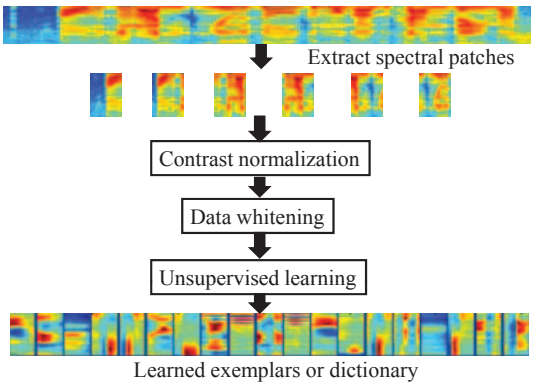

Fig. 1. Learning bag of exemplars of spectral patches.

2.1. Learning a bag of exemplars of spectral patches

Since the acoustic event pattern information is distributed in long-term temporal-frequency structure rather than in frame based short-term structure, we learn the pattern structure from spectral patches which is similar as we used before [13, 14]. The whole processing procedures are shown in Fig. 1. In this figure, the first step is spectral patch extraction from Mel band spectra. For simplicity, a uniform segment strategy is used for spectral patch selection in this study. All spectral patches are selected randomly from any time location in the spectrum of a training data set. The training feature vectors are created from all the training patches (by concatenating all frames in one patch as one feature vector). The second step as shown in Fig. 1 is contrast normalization. Similarly as used in image processing for local brightness and contrast normalization, each spectral patch is contrast normalized to remove the difference of the dynamic range caused by absolute density among patches. The contrast normalization is done as following:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - mean\left(\mathbf{x}\right)}{\sqrt{var\left(\mathbf{x}\right) + \varepsilon_c}}, \tag{1}$$

where $\mathbf{x}$ is the spectral patch vector (concatenating frame vectors to be a long vector), $mean(.)$ and $var(.)$ are the mean and variance operators, respectively. ε_c is a regularization parameter to make sure not to amplify the noise structure (set as 1 in this study). In exemplar learning, it is possible to learn only the correlation information if the data has strong correlation structure. This may result in weak representation power of the learned exemplars. In order to make exemplars span a good representative space with high order statistical structure, the data is further whitened for the third step as shown in Fig. 1. Principal component analysis (PCA) based whitening can be used to whiten the data. But the whitened data is represented in the PCA projection space. In order to make the whitened data as close as to the original input space, a zero-phase component analysis (ZCA) is used as follows [8, 9]:

$$\hat{\mathbf{x}} = \mathbf{V}\left(\mathbf{D} + \varepsilon_{\mathbf{w}}\mathbf{I}\right)^{-1/2}\mathbf{V}^T\tilde{\mathbf{x}}_{\mathbf{c}}, \tag{2}$$

where $\mathbf{V}$ and $\mathbf{D}$ are the eigen vector and eigen value matrix of the covariance matrix $\mathrm{cov}\left(\tilde{\mathbf{x}}_{\mathbf{c}}\right)$, $\tilde{\mathbf{x}}_{\mathbf{c}}$ is the zero centered vector of $\tilde{\mathbf{x}}$, $\varepsilon_{\mathbf{w}}$ is the regularization parameter in whitening which is used to reduce the effect of the eigen vectors with small eigen values (set as 0.1 in this study). The whitened spectral patches are used to learn the bag of exemplars. Many learning algorithms are available for different purpose, for example, sparse dictionary learning based on K-SVD [10], sparse dictionary learning based on projected gradient algorithm [11], or NMF algorithm. In this study, since our purpose is to learn representative spectral patches of the data, a simple k-means clustering which is widely used in vector quantization (VQ) is adopted to learn the codebook.

2.2. Sparse representation based on the learned bag of spectral exemplars

After learned the bag of exemplars, the signal can be represented based on the VQ. Traditionally, only one code vector is picked up to represent each feature vector by finding the most similar code in the codebook. In order to incorporate rich discriminative information from all the learned exemplars, the representation of one vector is based on the similarity measurement to all the learned exemplars as following:

$$\mathbf{d}_{\mathbf{y}} = \left[d_1, d_2, \cdots, d_i, \cdots, d_M\right]^T, \tag{3}$$

where M is the number of codewords. $\mathbf{d}_{\mathbf{y}}$ defines the similarity distance between feature vector $\mathbf{y}$ and learned exemplars. The similarity measurement can be defined with many types of metrics, for example, Euclidian distance, Gaussian kernel distance. In this study, an Euclidian distance based similarity metric as $d_i = \left\|\hat{\mathbf{x}} - \mathbf{c}^i\right\|_2$ is used where $\mathbf{c}^i$ is the ith exemplar in the learned codebook. Incorporating all exemplars in representation may make the representation not robust and less invariance to some distortions (e.g., noise or other distortions). We think of using only dominant exemplars which are with high similarity to the signal (small Euclidian distance) for feature representation, i.e., sparse representation based on similarity measurement as:

$$\begin{aligned} &\mathbf{y} = \left[y_1, y_2, \cdots, y_i, \cdots, y_M\right]^T \\ &y_i = \max\left(0, \lambda * \mathrm{mean}\left(\mathbf{d}_{\mathbf{y}}\right) - d_i\right), i = 1, 2, \cdots, M \end{aligned} \tag{4}$$

λ is sparsity control parameter. When $\lambda = 1$, it is the triangle VQ representation as proposed in [9]. The meaning of Eq. 4 is to make the activities as zeros when the distance measurements are larger than some ratio of an average threshold. This representation can reflect the data pattern similarity structure of the space expanded by exemplars with uncertainty across multiple dominant exemplars.

3. TRAINING CLASSIFIERS

The sparse representation is in a high dimensional space, which is not suitable to model using GMM, therefore the

SVM classifier is used in this study. For convenience of analysis, a linear SVM is used [12]. Suppose we have training data pairs as $(\mathbf{z}_i, l_i)$, with $i = 1, 2, ..., N$, where l_i is the label, and $\mathbf{z}_i$ is the sparse feature vector (in real implementation it is extended from sparse feature $\mathbf{y}_i$ for bias term). Multi-class SVMs (M) are built, and each SVM for each acoustic event is constructed as one-against-all with parameter $\mathbf{w}_j$ (the j-the SVM) as:

$$\underset{\mathbf{w}_j}{\text{minimize}} \sum_{i=1}^{N} \left(\max\left\{0, 1 - l_i\left(\mathbf{w}_j^T \mathbf{z}_i\right)\right\}\right)^2 + \alpha \left\|\mathbf{w}_j\right\|_2^2 \quad (5)$$

The classification can be done by picking up the one which gives the maximum value from all the SVMs as:

$$\hat{l} = \arg \max_{j \in \{1,2,\cdots M\}} \mathbf{w}_j^T \mathbf{z} \quad (6)$$

4. EXPERIMENTS

Our primary experiments were carried out on TED (technology, entertainment, and design) talks audio data. In the TED talks, besides speech, other acoustic events exist, for example, applause, laugh, and music events. In order to classify the underlying audio data streams to their event categories, manually labels were made. We chose 50 TED talks as training set, and 10 TED talks as testing set. On average, each TED talk has about 15 minutes audio data with 16kHz sampling rate. From the original manual event transcription, we collected nine event categories with semantic labels as {speech, applause, cough, laugh, audience, video, music, mix, other}. Among them, mix event is a collection of overlapped acoustic events, e.g., applause mixed with laugh. Other event is a collection of events we have not defined well in our application, such as some moving of chairs in a lecture, or natural environment sounds. As a detection task, performance evaluation metrics are related to false alarm rate and hitting rate. For audio data, these metrics can be frame based, event based or class-wise event based evaluation [1, 3]. In this study, frame based evaluation is used, i.e., frame based Rec (recall), Pre (precision) and F evaluation metrics are used which are the same as defined in [5].

4.1. Event detection based on the HMM

Since we have manual transcription of the event categories, it is natural to think of training HMM models for acoustic event detection just the same as used in ASR. For comparison purpose, we built an HMM event detection system based on HTK [15] (the results will be used for comparison analysis in the next section). Each event is modeled as a one state HMM (in this case, it is equivalent to GMM modeling), 3 states HMM with 16 GMMs for each state emission probability estimation (ergodic state transitions). Frame based 39 dimensional features including MFCC feature and log energy with their first and second order derivatives are used (CMN is applied for feature normalization). Considering the time duration difference of frame based and patch based representations, we smooth the label values to fit to the time window of patch based processing. The performance results are shown in table 1. In this table, HMM-1, HMM-2, and HMM-3 represent the HMM models with 1, 2 and 3 states (with output probability), respectively. From table 1, we found that one state HMM gave the best performance. We have thought to use more states to capture the long temporal statistical feature in HMM for event detection. However, from table 1, we found that simply adding more states in HMM degrades the performance.

Table 1. Event detection results based on HMM (%)

Model	Rec	Pre	F
HMM-1	74.95	75.46	75.20
HMM-3	70.57	71.06	70.81

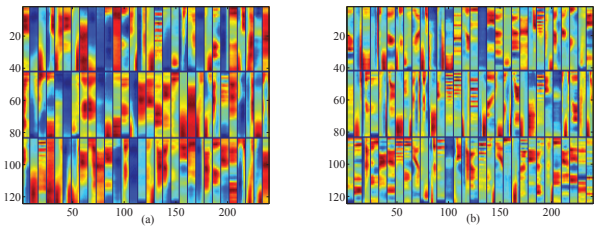

Fig. 2. Exemplars learned for patch size 7 frames without (a) and with (b) data whitening.

4.2. SVM with sparse representation model

There are many factors may affect the performance of the proposed representation. In this subsection we discuss the factors of data whitening, codebook size (CS) of exemplars, spectral patch size (PS) (i.e., how many frames to form one patch), and sparsity of the representation (via changing parameter λ in Eq. 4).

4.2.1. Data whitening

As we have discussed in section 2.1, data whitening makes the learning focus on high-order statistical structure (the second order statistical structure is removed by whitening). In this sense, the learned exemplars should span the feature space more uniformly which make the exemplars much more representative than those learned dominated by correlation structure. We show the learned exemplars of the spectral patches with and without whitening in Fig. 2. In this figure, each learned exemplar is a small patch with size of 40 frequency bands by 7 frames ((a) and (b)). From this figure, we can see that, after the data is whitened (panel (b)), the learned spectral patches show much more sharper local temporal-frequency structures (e.g., harmonic structure and frequency transitions). These structures can be regarded as

Table 2. Effect of whitening with CS = 128 (%). CS: code book size, PS: patch size.

	Rec	Pre	F
No (PS=7)	56.85	57.17	57.01
Whitening (PS=7)	67.30	67.70	67.50
No (PS=15)	57.85	60.21	60.03
Whitening (PS=15)	74.12	74.58	74.35

Table 3. Effect of codebook size (PS = 7) (%)

CS	64	128	256	512	1024
Rec	63.15	67.30	71.09	72.24	73.44
Pre	63.52	67.70	71.51	72.69	73.90
F	63.33	67.50	71.30	72.47	73.67

enhanced feature to represent acoustic event patterns. We did experiments to test the performance, and show the results in table 2 (the sparsity control parameter is fixed as $\lambda = 1$). From this table, we can see that with the whitening process, the detection performance is significantly improved. Therefore, in the following experiments, the data is with whitening process.

4.2.2. Codebook size of exemplars

Intuitively, a large number of exemplars can be much more accurate to represent the pattern acoustic space than a small number of exemplars. We did experiments to test the effect of increasing the codebook size of exemplars and show the results in table 3 (fixed $\lambda = 1$ in Eq. 4). From this table, we see a continuous improvement with increasing of the codebook size of exemplars. However, with increasing of the number of exemplars, it is possible that non-representative exemplars encoding noise structure may result in less invariance of the sparse representation. In the future, we will examine the robustness of the representation when the number of exemplars is increased to large values (e.g., more than several thousands).

4.2.3. Spectral patch size

As we have discussed that long temporal window for spectral patch should be helpful for catching rich event pattern information spanning beyond several frames. However, spectral patches with too long temporal window containing two or more events may bring large pattern confusion both in representation and model training. We did experiments to test the selection of spectral patch size and show results in tables 4 and 5 (fixed $\lambda = 1$ in Eq. 4). From these tables, we can see that increasing the spectral patch size could improve the performance. Nevertheless, the improvements became smaller when the spectral patch size became large, and the performance decreased when the size of spectral patch is increased beyond some large values.

Table 4. Effect of spectral patch size (CS = 256) (%)

PS	3	7	11	15	19	23
Rec	58.30	71.09	75.04	77.40	79.57	78.76
Pre	58.61	71.51	75.51	77.89	80.08	79.34
F	58.45	71.30	75.28	77.64	79.82	79.05

Table 5. Effect of spectral patch size (CS = 512) (%)

PS	3	7	11	15	19	23
Rec	60.63	72.24	75.92	78.98	80.44	79.82
Pre	60.96	72.69	76.40	79.49	80.95	80.52
F	60.79	72.47	76.16	79.24	80.69	80.17

Table 6. Effect of representation sparsity (CS = 256) (%)

λ	0.8	1.0	1.2	1.4	1.6	1.8
Rec	54.08	75.04	70.85	68.44	69.20	69.21
Pre	54.36	75.50	71.28	68.86	69.62	69.63
F	54.22	75.28	71.06	68.65	69.41	69.42

4.2.4. Representation sparsity

The representation sparseness can be controlled by varying λ in Eq. 4. It can adjust the tradeoff between representation accuracy and robustness. We did experiments with different λ values, and showed the results in tables 6 and 7. From these tables, we can confirm that the sparsity should be chosen in a range to make the representation with good representation accuracy while keeping robustness or discriminability.

5. CONCLUSION AND DISCUSSION

In this study, we learned the spectral patch exemplars by using a k-means clustering algorithm on whitened spectral patches. Considering the representation accuracy and robustness, a sparse representation based on the learned spectral exemplars was extracted. With the sparse representation, an SVM classification system was built for acoustic event detection. Our experiments showed that the sparse representation with an SVM classifier can outperform the traditional HMM modeling on frame based representations (refer to tables 1, 4 and 5).

Recently, deep learning is widely used for speech and vision processing tasks for feature extraction and classification [16]. The essential thing done by the deep learning is to extract powerful representative features [17]. As shown in several studies, if the raw data is with a proper preprocessing, good features that are competitive to those explored by deep learning can be obtained by a single layer or shallow learning [9]. The philosophy in all these feature learning is to disentangle the underlying factors to represent patterns [17]. Both the deep learning and the work in this study try to explore the factors in spectral patches with uniformed segments. However, the factors may be distributed in non-uniformed spectral segments. Finding such kinds of non-uniform acoustic representation remains as our future work.

Table 7. Effect of representation sparsity (CS = 512) (%)

λ	0.8	1.0	1.2	1.4	1.6	1.8
Rec	65.33	75.92	73.12	71.91	70.34	70.07
Pre	65.70	76.40	73.57	72.35	70.78	70.50
F	65.51	76.16	73.34	72.13	70.56	70.28

6. REFERENCES

[1] D. Giannoulisy, E. Benetosx, D. Stowelly, M. Rossignolz, M. Lagrangez and M. Plumbley, "Detection and Classification of Acoustic Scenes and Events: an IEEE AASP Challenge," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[2] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 1, pp. 1-13, 2013.

[3] X. Zhuang, X. Zhou, M. A. Hasegawa-johnson, T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543-1551, 2010.

[4] C. Zieger, "An HMM based system for acoustic event detection," *Multimodel technologies for perception of humans*, pp. 338-344, 2008.

[5] A. Temko, C. Nadeu, and J. I. Biel, "Acoustic Event Detection: SVM-Based System and Evaluation Setup in CLEAR'07," *Multimodel technologies for perception of humans*, pp. 354-363, 2008.

[6] Z. Huang, Y. Cheng, K. Li, V. Hautamaki, C. Lee, "A Blind Segmentation Approach to Acoustic Event Detection Based on I-Vector," in *Proc. Interspeech*, pp. 2282–2286, 2013.

[7] C. V. Cotton, D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Proc. WASPAA*, pp. 69-72, 2010.

[8] A. J. Bell, T. J. Sejonwski, "The gIndependent Componentsh of Natural Scenes are Edge Filters," Vision Res., vol. 37, no. 23, pp. 3327-3338, 1997.

[9] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. the 14-th International Conference on AI and Statistics*, 215-223, 2011.

[10] M. Aharon, M. Elad and A. Bruckstein, K-SVD,"An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol 54, no. 11, pp. 4311-4322, 2006.

[11] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online Dictionary Learning for Sparse Coding," *International Conference on Machine Learning*, Montreal, Canada, 2009

[12] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.

[13] X. Lu, S.Matsuda, C. Hori, H. Kashioka, "Speech restoration based on deep learning autoencoder with layer-wised learning," INTERSPEECH, Portland, Oregon, Sept., 2012.

[14] X. Lu, Y. Tsao, S. Matsuda, C. Hori, "Speech Enhancement Based on Deep Denoising Autoencoder," INTERSPEECH, Aug. 26, 2013, Lyon, France.

[15] The HTK Book (version 3.2), (2002). Cambridge University Engineering Department.

[16] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

[17] Y. Bengio, and A. Courville, "Deep Learning of Representations," in *Handbook on Neural Information Processing*, Springer, Berlin Heidelberg, 2013.