Multimodal Sparse Coding for Event Detection

Youngjune Gwon William Campbell Kevin Brady Douglas Sturim
MIT Lincoln Laboratory, Lexington, MA 02420, USA

Miriam Cha H. T. Kung
Harvard University, Cambridge, MA 02138, USA

Abstract

Unsupervised feature learning methods have proven effective for classification tasks based on single modality. We present multimodal sparse coding for learning feature representations shared across multiple modalities. The shared representations are applied to multimedia event detection (MED) and evaluated in comparison to unimodal counterparts, as well as other feature learning methods such as sparse autoencoder and RBM. We report the cross-validated classification accuracy and mean average precision of the MED system trained on features learned from our unimodal and multimodal settings for the TRECVID MED 2014 dataset.

1 Introduction

Multimedia Event Detection (MED) aims to identify complex activities occurring at specific place and time, involving various interactions of human actions and objects. MED is considered more difficult than concept analysis such as action recognition and has received significant attention in computer vision and machine learning research. In this paper, we propose the use of sparse coding for multimodal feature learning in the context of MED. Originally proposed to explain neurons encoding sensory information [6], sparse coding provides an unsupervised method to learn basis vectors for efficient data representation. More recently, sparse coding has been used to model the relationship between correlated data sources. By jointly training dictionaries with audio and video tracks from the same multimedia clip, we can force the two modalities to share a similar sparse representation whose benefit includes robust detection and cross-modality retrieval.

In the next section, we will describe audio-video feature learning in various unimodal and multimodal settings for sparse coding. We then present our experiments with TRECVID MED dataset. We will discuss the empirical results, compare them to other methods, and conclude.

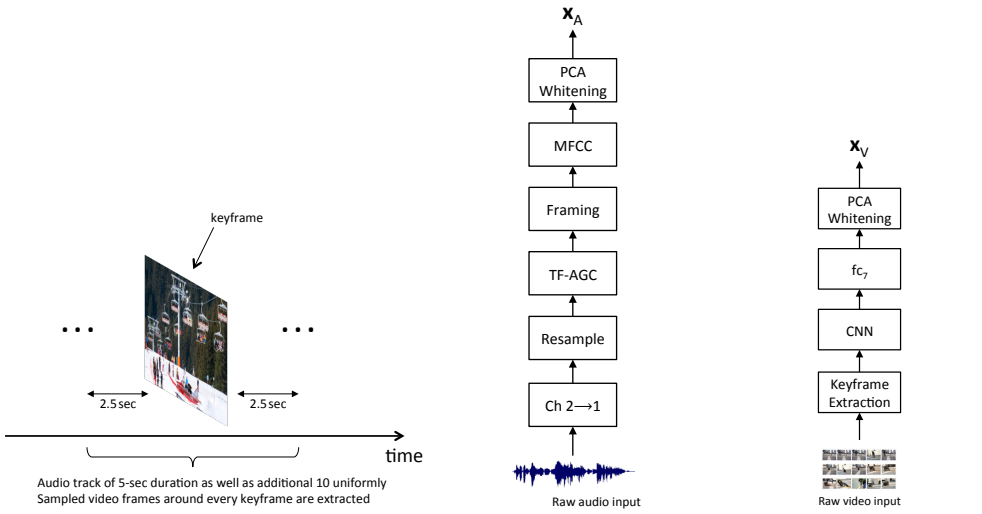
2 Audio-video Feature Learning

In summary, our approach is to build feature vectors by sparse coding on the low-level audio and video features. Multiple feature vectors (*i.e.*, sparse codes) are aggregated via max pooling. The resulting, pooled feature vectors can scale to file level, and we use them to train an array of classifiers for MED.

Distribution A: Public Release.

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002.

Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.



(a) Keyframe extraction (b) Audio preprocessing (c) Video preprocessing

Figure 1: Preprocessing audio and video data from multimedia clip

2.1 Low-level feature extraction and preprocessing

We begin by locating the keyframes of a given multimedia clip. We apply a simple two-pass algorithm that computes color histogram difference of any two successive frames and determines a keyframe candidate based on the threshold calculated on the mean and standard deviation of the histogram differences. We examine the number of different colors present in the keyframe candidates and discard the ones with less than 26 colors. This ensures that our keyframes are not all-black or all-white blank images.

Around each keyframe, we extract 5-sec audio data and additional 10 uniformly sampled video frames within the duration as illustrated in Figure 1a. If extracted audio is stereo, we take only its left channel. The audio waveform is resampled to 22.05 kHz and regularized by the time-frequency automatic gain control (TF-AGC) to balance the energy in sub-bands. We form audio frames using a 46-msec Hann window with 50% overlap between successive frames for smoothing. For each frame, we compute 16 the Mel-frequency cepstral coefficients (MFCCs) as the low-level audio feature. In addition, we append 16 delta cepstral and 16 delta-delta cepstral coefficients, which make our low-level audio feature vectors 48 dimensional. Finally, we apply PCA whitening before unsupervised learning. The complete audio preprocessing steps are described in Figure 1b.

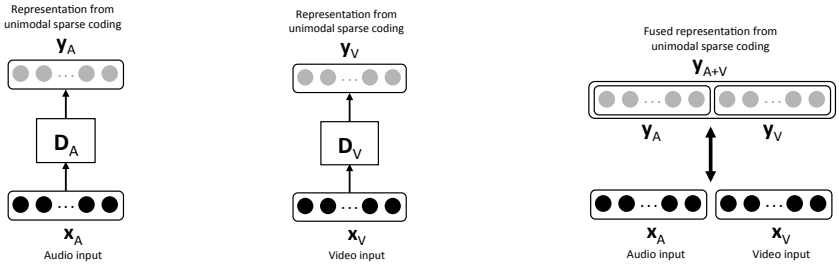
For video preprocessing, we take a deep learning approach. We have tried out pretrained convolutional neural network (CNN) models and ended up choosing `VGG_ILSVRC_19_layers`, the 19-layer model by University of Oxford's Visual Geometry Group (VGG) [7] for the ImageNet Large-scale Visual Recognition Challenge (ILSVRC). As depicted in Figure 1c, we run the CNN feed-forward passes with the extracted video frames. For each video frame, we take 4,096-dimensional hidden activation from fc_7, the highest hidden layer before the final ReLU (*i.e.*, the rectification non-linearity). By PCA whitening, we reduce the dimensionality to 128.

2.2 High-level feature modeling via sparse coding

We use sparse coding to model high-level features that can train classifiers for event detection.

Unimodal feature learning. A straightforward approach for sparse coding with two heterogeneous data modalities is to learn a *separate* dictionary of basis vectors for each modality. Figure 2 depicts unimodal sparse coding schemes. Recall the preprocessed audio and video input vectors $\mathbf{x}_\mathrm{A}$ and $\mathbf{x}_\mathrm{V}$. Audio sparse coding is expressed by

$$\min_{\mathbf{D}_\mathrm{A}, \mathbf{y}_\mathrm{A}^{(i)}} \sum_{i=1}^{n_\mathrm{A}} \|\mathbf{x}_\mathrm{A}^{(i)} - \mathbf{D}_\mathrm{A}\mathbf{y}_\mathrm{A}^{(i)}\|_2^2 + \lambda\|\mathbf{y}_\mathrm{A}^{(i)}\|_1 \tag{1}$$



(a) Audio sparse coding (b) Video sparse coding (c) Feature union

Figure 2: Unimodal sparse coding and feature union

where we feed n_{A} unlabeled audio examples to learn the unimodal dictionary $\mathbf{D}_{\mathrm{A}}$ and sparse codes $\mathbf{y}_{\mathrm{A}}^{(i)}$ simultaneously. (We denote $\mathbf{x}_{\mathrm{A}}^{(i)}$ the ith training example for audio.) Similarly, using n_{V} unlabeled video examples, we learn

$$\min_{\mathbf{D}_{\mathrm{V}},\mathbf{y}_{\mathrm{V}}^{(i)}} \sum_{i=1}^{n_{\mathrm{V}}} \|\mathbf{x}_{\mathrm{V}}^{(i)} - \mathbf{D}_{\mathrm{V}}\mathbf{y}_{\mathrm{V}}^{(i)}\|_2^2 + \lambda\|\mathbf{y}_{\mathrm{V}}^{(i)}\|_1. \tag{2}$$

We can form $\mathbf{y}_{\mathrm{A+V}} = [\mathbf{y}_{\mathrm{A}}\ \mathbf{y}_{\mathrm{V}}]^\top$, a union of the audio and video feature vectors from unimodal sparse coding illustrated in Figure 2c.

Multimodal feature learning. The feature union $\mathbf{y}_{\mathrm{A+V}}$ encapsulates both audio and video sparse codes. However, the training is done in a parallel, unimodal fashion such that sparse coding dictionary for each modality is learned independently of the other. To remedy the lack of joint learning, we propose a multimodal sparse coding scheme described in Figure 3a. We use the joint sparse coding technique used in image super-resolution [9]

$$\min_{\mathbf{D}_{\mathrm{AV}},\mathbf{y}_{\mathrm{AV}}^{(i)}} \sum_{i=1}^{n} \|\mathbf{x}_{\mathrm{AV}}^{(i)} - \mathbf{D}_{\mathrm{AV}}\mathbf{y}_{\mathrm{AV}}^{(i)}\|_2^2 + \lambda'\|\mathbf{y}_{\mathrm{AV}}^{(i)}\|_1. \tag{3}$$

Here, we feed the concatenated audio-video input vector $\mathbf{x}_{\mathrm{AV}}^{(i)} = [\frac{1}{\sqrt{N_{\mathrm{A}}}}\mathbf{x}_{\mathrm{A}}^{(i)}\ \frac{1}{\sqrt{N_{\mathrm{V}}}}\mathbf{x}_{\mathrm{V}}^{(i)}]^\top$, where N_{A} and N_{V} are dimensionalities of $\mathbf{x}_{\mathrm{A}}$ and $\mathbf{x}_{\mathrm{V}}$, respectively. As an interesting property, we can decompose the jointly learned dictionary $\mathbf{D}_{\mathrm{AV}} = [\frac{1}{\sqrt{N_{\mathrm{A}}}}\mathbf{D}_{\mathrm{AV-A}}\ \frac{1}{\sqrt{N_{\mathrm{V}}}}\mathbf{D}_{\mathrm{AV-V}}]^\top$ to perform the following audio-only and video-only sparse coding

$$\min_{\mathbf{D}_{\mathrm{AV-A}},\mathbf{y}_{\mathrm{AV-A}}^{(i)}} \sum_{i=1}^{n_{\mathrm{A}}} \|\mathbf{x}_{\mathrm{AV-A}}^{(i)} - \mathbf{D}_{\mathrm{AV-A}}\mathbf{y}_{\mathrm{AV-A}}^{(i)}\|_2^2 + \lambda''\|\mathbf{y}_{\mathrm{AV-A}}^{(i)}\|_1, \tag{4}$$

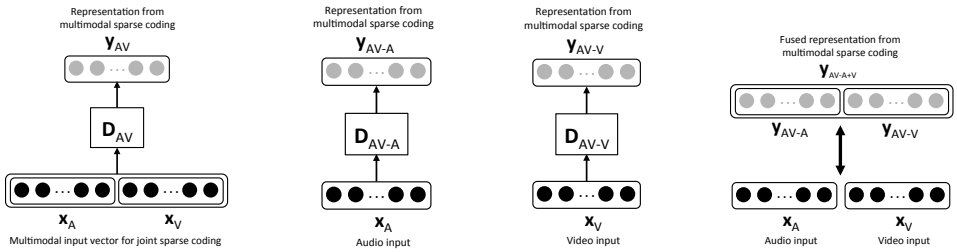
$$\min_{\mathbf{D}_{\mathrm{AV-V}},\mathbf{y}_{\mathrm{AV-V}}^{(i)}} \sum_{i=1}^{n_{\mathrm{V}}} \|\mathbf{x}_{\mathrm{AV-V}}^{(i)} - \mathbf{D}_{\mathrm{AV-V}}\mathbf{y}_{\mathrm{AV-V}}^{(i)}\|_2^2 + \lambda''\|\mathbf{y}_{\mathrm{AV-V}}^{(i)}\|_1. \tag{5}$$

In principle, joint sparse coding via Eq. (3) combines the objectives of Eqs. (4) and (5), forcing the sparse codes $\mathbf{y}_{\mathrm{AV-A}}^{(i)}$ and $\mathbf{y}_{\mathrm{AV-V}}^{(i)}$ to share the same representation. Ideally, we could have $\mathbf{y}_{\mathrm{AV}}^{(i)} = \mathbf{y}_{\mathrm{AV-A}}^{(i)} = \mathbf{y}_{\mathrm{AV-V}}^{(i)}$, although empirical values determined by the three different optimizations differ in reality. Feature formation possibilities on multimodal sparse coding are explained in Figure 3.

3 Evaluation

3.1 Dataset, task, and experiments

We use the TRECVID MED 2014 dataset [1] to evaluate our schemes. There are 20 event classes E021–E040 for TRECVID MED 2014 with event names such as "Bike trick," "Dog show," and "Marriage proposal." We consider the event detection and retrieval tasks using the 10Ex and 100Ex data directories, where 10Ex includes 10 multimedia examples per event, and 100 examples for



(a) Joint sparse coding (b) Audio only (c) Video only (d) Union

Figure 3: Multimodal sparse coding and feature formation possibilities

Table 1: TRECVID MED 2014 performance comparisons for unimodal and multimodal sparse coding schemes (see Figures 2 and 3 for details on feature formation).

	Unimodal			Multimodal			
	Audio SC	Video SC	Union	Audio-only	Video-only	Joint	Union
Mean accuracy (%)	75	86	**89**	62	87	90	**91**
mAP (%)	20.0	33.1	**34.8**	27.4	35.3	28.1	**37.9**

100Ex. We compute feature vectors on the unimodal and multimodal sparse coding schemes in Figures 2 and 3.

We use the number of basis vectors $K = 512$ same for all of the sparse coding dictionaries $\mathbf{D}_{\mathrm{A}}$, $\mathbf{D}_{\mathrm{V}}$, and $\mathbf{D}_{\mathrm{AV}}$. Using the max-pooled sparse codes, we train linear, 1-vs-all SVM classifiers for each event whose hyper-parameters are determined by 5-fold cross-validation. We have used the INRIA SPAMS (SPArse Modeling Software) [2] package for sparse coding, VOICEBOX Speech Processing Toolkit [3], MatConvNet [8] to drive the pretrained deep CNN models, and LIBSVM [4].

Our evaluation is based on the following experiments.

1. Cross-validation on 10Ex
2. Cross-validation on 100Ex
3. Train with 10Ex and test on 100Ex

For evaluation metrics, we adopt classification accuracy and mean average precision (mAP). We compute mAP to evaluate the event detection and retrieval performance according to the NIST standard.

3.2 Other feature learning methods for comparison

In addition, we consider other unsupervised methods to learn audio-video features for comparison. We report the results for Restricted Boltzmann Machine (RBM) and autoencoder neural network (AE) for both unimodal and multimodal cases. In particular, we adopt the shallow bimodal pretraining model [5] for unsupervised feature learning. With a fixed hidden layer size of 512 (*i.e.*, same as the dimensionality of sparse codes), we apply the target sparsity 0.15 for both RBM and AE. The resulting hidden activations are similarly max-pooled and aggregated to file-level feature vectors before SVM.

3.3 Results

Table 1 presents the classification accuracy and mAP performance of unimodal and multimodal sparse coding schemes on TRECVID MED 2014 10Ex. We observe that the union of pooled audio and video feature vectors perform the best for both cases. As our union operation concatenates the two feature vectors, the resulting descriptor is doubled in dimensionality. Joint feature vector is an economical way of combining both the audio and video features as it maintains the same dimensionality as audio-only or video-only. Despite superior classification accuracy, its mAP for retrieval is found worse than video-only.

4 Conclusion

We have presented multimodal sparse coding for MED. Our approach can build joint sparse feature vectors learned from different modalities and scale to file-level descriptors suitable for training classifiers in a MED system. Using the TRECVID MED 2014 dataset, we have empirically validated our approach and achieved promising performance measured in accuracy and precision metrics recommended by the NIST standard. Our future work includes an integration with more features, training for larger event coverage, and finetuning.

References

[1] 2014 TRECVID Multimedia Event Detection & Multimedia Event Recounting Tracks. `http://nist.gov/itl/iad/mig/med14.cfm`.

[2] SPArse Modeling Software. `http://spams-devel.gforge.inria.fr/`.

[3] VOICEBOX: Speech Processing Toolbox for MATLAB. `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[5] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal Deep Learning. In *ICML*, 2011.

[6] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision research*, 37(23):3311–3325, 1997.

[7] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[8] A. Vedaldi and K. Lenc. MatConvNet – Convolutional Neural Networks for MATLAB. 2015.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image Super-Resolution via Sparse Representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, Nov 2010.