INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Subspace Pooling Based Temporal Features Extraction For Audio Event Recognition

Qiuying Shi, Hui Luo, Jiqing Han

School of Computing Science and Technology,
Harbin Institute of Technology, Harbin, China

sqyshiqiuying@gmail.com, luohui0216@outlook.com, jqhan@hit.edu.cn

Abstract

Currently, most popular methods of Audio Event Recognition (AER) firstly split audio event signals into multiple short segments, then the features of these segments are pooled for recognition. However, the temporal features between segments, which highly affect the semantic representation of signals, are usually discarded in the above pooling step. Thus, how to introduce the temporal features to the pooling step requires further investigation. Unfortunately, on the one hand, only a few studies have been conducted towards solving this problem so far. On the other hand, the effective temporal features should not only capture the temporal dynamics but also have the signal reconstruction ability, while most of the above studies mainly focus on the former but ignore the latter. In addition, the effective features of high-dimensional original signals usually inhabit a low-dimensional subspace. Therefore, we propose two novel pooling based methods which try to consider both the temporal dynamics and signal reconstruction ability of temporal features in the low-dimensional subspace. The proposed methods are evaluated on the AudioEvent database, and experimental results show that our methods can outperform most of the typical methods.

Index Terms: acoustic event recognition, temporal features extraction, subspace pooling, non-negative, sparse

1. Introduction

The task of attributing a semantic label to a specific audio event signal is commonly referred as Audio Event Recognition (AER) [1]. One important issue in AER is to extract temporal features of time-sensitive signals. However, most of conventional methods [2–6] are designed to extract features (e.g., Mel-Frequency Cepstral Coefficients (MFCCs) [3–6]) for each single frame which is too short to contain temporal characteristics of real-life signals. Generally, temporal features should be captured based on multiple frames (i.e., a segment of signals).

To obtain segment-level features, pooling methods, which can aggregate the features of multiple segments, become widely used in AER. As a typical kind of these methods, Bag-of-Audio-Words (BoAW)-based methods [7–10] are studied for describing segments by histograms. Besides, another kind of pooling methods, which can compute the maximum or average of segments, is often adopted by most of Deep Neural Networks (DNN)-based methods [11–14], such as Convolutional Neural Networks (CNN)-, Convolutional Long Short-Term Memory (ConvLSTM)-based methods, etc. Although these existing pooling methods can achieve fairly good performance, they usually discard the temporal features between segments which highly affect the semantic representation of signals. Since the above semantic representation highly affects the performance of AER, it is necessary to further explore a new pooling method which can capture the temporal features between segments.

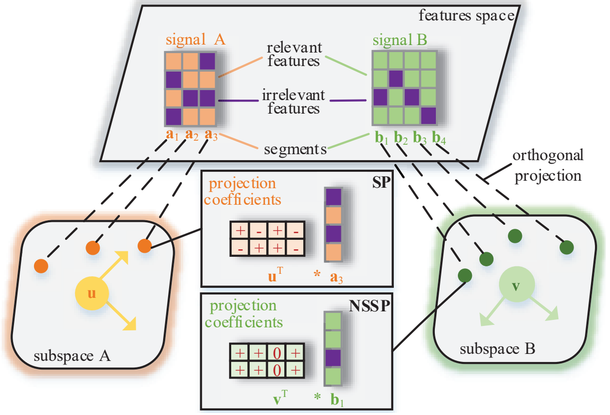

Figure 1: *An illustration of the proposed methods, i.e., Subspace Pooling (SP) and Non-negative Sparse Subspace Pooling (NSSP). For every signal, both methods extract the temporal features between segments by learning a low-dimensional subspace. And the basis of subspaces (e.g.,* $\mathbf{u}$ *or* $\mathbf{v}$*) is used as an efficient representation for each signal. In addition, the NSSP further introduces non-negative and sparse constraints to the basis based on the SP for robust AER.*

To this end, a Rank Pooling (RP) method is studied for capturing temporal dynamics between segments which is a vital part of the temporal features [15]. However, since the temporal features should not only capture the temporal dynamics but also have the signal reconstruction ability, the RP which ignores the latter is inadequate for effectively representing the temporal features. To address this issue, a variant of the BoAW is further designed to consider the signal reconstruction ability before capturing temporal dynamics with the RP [15]. However, these two vital parts of the temporal features are considered independently which reduce the completeness of the temporal features. Thus, it is necessary to jointly consider these two parts.

In this paper, two efficient pooling methods are proposed to capture the temporal features between segments, which are shown in Figure 1. The main idea of these methods is to effectively consider both the temporal dynamics and signal reconstruction ability of the temporal features. Since the effective features of high-dimension original signals usually inhabit a low-dimensional subspace [16], we try to extract the temporal features in the low-dimensional subspace.

The main contributions of this paper are summarized as: First, we propose a new pooling strategy for effectively extracting the temporal features by learning a low-dimensional subspace. Second, we first introduce the signal reconstruction ability of the temporal features to the pooling methods by mini-

http://dx.doi.org/10.21437/Interspeech.2019-2047

mizing the reconstruction error scheme. Third, we propose two efficient solutions for our proposed methods respectively.

2. Proposed Methods

2.1. Subspace Pooling method

The main purpose of this paper is to design new pooling methods which can extract temporal features between segments. And we hope that they can be realized by learning a low-dimensional subspace which perfectly encodes two parts of the temporal features, i.e., the temporal dynamics and signal reconstruction ability. In order to clearly describe our proposed pooling methods which focus on these two vital parts, the joint learning process of the low-dimension subspace will be given next, which firstly considers the signal reconstruction ability and then captures the temporal dynamics.

First, for considering the signal reconstruction ability, one of the most widely adopted way for learning a subspace is to use the reconstruction error minimization scheme. More specifically, the subspace can be learned by minimizing the reconstruction error between the original signal and corresponding orthogonal projections [17], and its basis is:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \|\mathbf{X} - \mathbf{u}(\mathbf{u}^{\mathrm{T}}\mathbf{u})^{-1}\mathbf{u}^{\mathrm{T}}\mathbf{X}\|_{\mathrm{F}}^2 \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ is a feature matrix for a randomly selected signal which contains N segments, and $\mathbf{x}_n \in \mathbb{R}^D (n = 1, \ldots, N)$ is D-dimensional segment-level features for a segment; $\mathbf{u} \in \mathbb{R}^{D \times d}$ is the basis of a d-dimensional subspace $\mathbb{S}$; $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm [18]. And how to extract these segment-level features will be described in section 3. Furthermore, for a signal which is composed by multiple segments, a pooling method always can aggregate these segment-level features into a single signal-level features. Inspired by this idea, we reduce the dimension of subspace $\mathbb{S}$ to one and reformulate (1) as:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \|\mathbf{X} - \mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{X}\|_{\mathrm{F}}^2 \tag{2}$$

Clearly that the signal reconstruction ability is well ensured in the one-dimensional subspace learned by (2), but how to further capture the temporal dynamics requires more investigation.

Then, to this end, we reconsider the target of (2), i.e., reconstructing $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from the corresponding orthogonal projections $\mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{x}_1, \ldots, \mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{x}_N$. And each orthogonal projection $\mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{x}_i$ can be regarded as scaling the basis $\mathbf{u}$ by projection coefficients $\mathbf{u}^{\mathrm{T}}\mathbf{x}_i$. It is easy to find that in the orthogonal projections, the basis $\mathbf{u}$ maintains unchanged during the evolution of N segments and the projection coefficients $\mathbf{u}^{\mathrm{T}}\mathbf{x}_1, \ldots, \mathbf{u}^{\mathrm{T}}\mathbf{x}_N$ vary significantly from one segment to another. Thus, it is reasonable to consider that the temporal dynamics between N segments can be reflected by the projection coefficients. In addition, the evolution of N segments can also be described by the principle component of $\mathbf{X}$ which can be calculated by Singular Value Decomposition (SVD) [19]. Therefore, one way to learn the one-dimensional subspace, which can further capture the temporal dynamics, is to build connections between the projection coefficients and principle components of $\mathbf{X}$.

In detail, $\mathbf{X}$ is decomposed via truncated SVD into three parts as:

$$\mathbf{X} = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^{\mathrm{T}} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{D \times d}$ contains left-singular vectors of $\mathbf{X}$, $\mathbf{B}^{\mathrm{T}} \in \mathbb{R}^{d \times T}$ includes right-singular vectors and $\mathbf{\Sigma}$ is made of singular values. Since the vectors in $\mathbf{B}^{\mathrm{T}}$ can reflect the temporal evolution of $\mathbf{X}$, thus, the projection coefficients $\mathbf{u}^{\mathrm{T}}\mathbf{X}$ should satisfy:

$$\mathbf{u}^{\mathrm{T}}\mathbf{X} = \mathbf{B}^{\mathrm{T}} \tag{4}$$

Finally, the basis $\mathbf{u}$ can be derived as:

$$\mathbf{u} = pinv(\mathbf{X})^{\mathrm{T}}\mathbf{B} \tag{5}$$

where $pinv(\cdot)$ is the Moore-Penrose pseudo-inverse [20].

We name the above method as Subspace Pooling (SP). In the SP, the projection coefficients are required to reflect the temporal dynamics which is a vital part of temporal features. And since the temporal features highly affects the performance of AER, thus, it is necessary to further investigate the above coefficients for getting more effective temporal features.

2.2. Non-negative Sparse Subspace Pooling method

The projection coefficients $\mathbf{u}^{\mathrm{T}}\mathbf{X}$ can also be regarded as linear combinations between the basis $\mathbf{u}$ and different feature dimensions of $\mathbf{X}$. And the combination coefficients, i.e., $\mathbf{u}$, indicate how necessary of each dimension for keeping the effective temporal features. Thus, if there exist irrelative dimensions in $\mathbf{X}$, $\mathbf{u}$ has to discard them for robust recognition. To this end, a sparse constraint is employed on $\mathbf{u}$. In addition, on the one hand, the influence of relative dimensions should be enlarged. On the other hand, since the basis $\mathbf{u} \in \mathbb{R}^{D \times d}$ is composed of multiple elements $\mathbf{u}_i (i = 1, \ldots, d)$. Thus, each projection of $\mathbf{x}_n$ onto $\mathbf{u}_i$ is a representation of one part in original $\mathbf{x}_n$. And the additive combinations of d parts is not only a reasonable way but also a guarantee of reconstructing $\mathbf{x}_n$. Hence, a non-negative constraint should be further employed on $\mathbf{u}$. By introducing the sparse and non-negative constraints to the SP, its objective function can be reformulated as:

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}} \|\mathbf{X} - \mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{X}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{u}\|_1 \\ \text{s.t.} &\quad \mathbf{u} \geq \mathbf{0} \end{aligned} \tag{6}$$

where λ is a parameter to control the effect of the sparse constraint and $\| \cdot \|_1$ is the ℓ_1 norm [18].

We name the above method as Non-negative Sparse Subspace Pooling (NSSP). Although our NSSP provides several advantages, it leads to a more difficult optimization problem with constraints. Hence, it is necessary to give an efficient method to optimize the NSSP.

To this end, an alternating direction optimization scheme is also proposed. It first introduces auxiliary variables (i.e., $\mathbf{z}, \mathbf{h}$ in this case) to make the optimization of (6) separable, which can be formulated as:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{z},\mathbf{h}} \|\mathbf{X} - \mathbf{u}\mathbf{z}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{h}\|_1 \\ &\text{s.t. } \mathbf{u}^{\mathrm{T}}\mathbf{X} - \mathbf{z} = \mathbf{0}, \mathbf{u} - \mathbf{h} = \mathbf{0}, \mathbf{h} \geq \mathbf{0} \end{aligned} \tag{7}$$

Then the augmented Lagrangian function [21] of (7) is:

$$\begin{aligned} &L_\rho(\mathbf{u}, \mathbf{z}, \mathbf{h}, \mathbf{y}_1, \mathbf{y}_2) \\ &= \|\mathbf{X} - \mathbf{u}\mathbf{z}\|_{\mathrm{F}}^2 + \lambda\|\mathbf{h}\|_1 + \frac{\rho}{2}\|\mathbf{u}^{\mathrm{T}}\mathbf{X} - \mathbf{z} + \frac{\mathbf{y}_1}{\rho}\|_2^2 \\ &+ \frac{\rho}{2}\|\mathbf{u} - \mathbf{h} + \frac{\mathbf{y}_2}{\rho}\|_2^2 - \frac{1}{2\rho}(\|\mathbf{y}_1\|_2^2 + \|\mathbf{y}_2\|_2^2) \end{aligned} \tag{8}$$

where $\mathbf{y}_1$, $\mathbf{y}_2$ are Lagrange multipliers, ρ is a penalty parameter and $\| \cdot \|_2$ is the ℓ_2 norm [18]. The (8) can be iteratively optimized as three subproblems with respect to $\mathbf{u}, \mathbf{z}$ and $\mathbf{h}$ respectively.

1) The subproblem for updating variable $\mathbf{u}$ with fixed variables $\mathbf{z}$ and $\mathbf{h}$ is:

$$\begin{aligned}\mathbf{u}^{k+1} &= \arg\min_{\mathbf{u}} \ \|\mathbf{X} - \mathbf{u}\mathbf{z}^k\|_{\mathrm{F}}^2 \\ &+ \frac{\rho_k}{2}(\|\mathbf{u}^{\mathrm{T}}\mathbf{X} - \mathbf{z}^k + \frac{\mathbf{y}_1^k}{\rho_k}\|_2^2 + \|\mathbf{u} - \mathbf{h}^k + \frac{\mathbf{y}_2^k}{\rho_k}\|_2^2)\end{aligned} \tag{9}$$

where, k is the index of iteration. Then, (9) is reformulated by replacing the original quadratic term with a first order approximation and a proximal term [21] as:

$$\begin{aligned}\mathbf{u}^{k+1} &= \arg\min_{\mathbf{u}} \ \|\mathbf{X} - \mathbf{u}\mathbf{z}^k\|_{\mathrm{F}}^2 \\ &+ \langle \bigtriangledown_{\mathbf{u}^k}\mathbf{\Psi}(\mathbf{u}^k, \mathbf{z}^k, \mathbf{h}^k, \mathbf{y}_1^k, \mathbf{y}_2^k, \rho_k), \mathbf{u} - \mathbf{u}^k\rangle \\ &+ \frac{\eta\rho_k}{2}\|\mathbf{u} - \mathbf{u}^k\|_2^2\end{aligned} \tag{10}$$

where $\eta > 0$, $\bigtriangledown_{\mathbf{u}^k}\mathbf{\Psi}$ is the partial derivatives of function $\mathbf{\Psi}$ with respect to variable $\mathbf{u}^k$ and $\langle\cdot,\cdot\rangle$ is the inner product. More specifically, the function $\mathbf{\Psi}$ is defined as :

$$\begin{aligned}&\mathbf{\Psi}(\mathbf{u}, \mathbf{z}, \mathbf{h}, \mathbf{y}_1, \mathbf{y}_2, \rho_k) \\ &= \frac{\rho_k}{2}(\|\mathbf{u}^{\mathrm{T}}\mathbf{X} - \mathbf{z} + \frac{\mathbf{y}_1}{\rho_k}\|_2^2 + \|\mathbf{u} - \mathbf{h} + \frac{\mathbf{y}_2}{\rho_k}\|_2^2)\end{aligned} \tag{11}$$

By setting the derivatives of (11) with respect to $\mathbf{u}$ to zeros, the update scheme for variable $\mathbf{u}$ can be derived as:

$$\begin{aligned}\mathbf{u}^{k+1} &= \frac{\rho_k((\eta - 1)\mathbf{u}^k + \mathbf{h}^k) - \mathbf{y}_2^k}{2\mathbf{z}^k(\mathbf{z}^k)^{\mathrm{T}} + \eta\rho_k} \\ &+ \frac{(2 + \rho_k)\mathbf{X}(\mathbf{z}^k)^{\mathrm{T}} - \rho_k\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{u}^k - \mathbf{X}(\mathbf{y}_1^k)^{\mathrm{T}}}{2\mathbf{z}(\mathbf{z}^k)^{\mathrm{T}} + \eta\rho_k}\end{aligned} \tag{12}$$

2) The subproblem for updating variable $\mathbf{z}$ with fixed variables $\mathbf{u}$ and $\mathbf{h}$ is:

$$\begin{aligned}\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \ \|\mathbf{X} - \mathbf{u}^{k+1}\mathbf{z}\|_{\mathrm{F}}^2 \\ &+ \frac{\rho_k}{2}\|(\mathbf{u}^{k+1})^{\mathrm{T}}\mathbf{X} - \mathbf{z} + \frac{\mathbf{y}_1^k}{\rho_k}\|_2^2\end{aligned} \tag{13}$$

By setting the derivatives of (13) with respect to $\mathbf{z}$ to zeros, the update scheme for $\mathbf{z}$ is:

$$\mathbf{z}^{k+1} = \frac{(2 + \rho_k)(\mathbf{u}^{k+1})^{\mathrm{T}}\mathbf{X} + \mathbf{y}_1^k}{2(\mathbf{u}^{k+1})^{\mathrm{T}}\mathbf{u}^{k+1} + \rho_k} \tag{14}$$

3) The subproblem for updating variable $\mathbf{h}$ with fixed variables $\mathbf{u}$ and $\mathbf{z}$ is:

$$\mathbf{h}^{k+1} = \arg\min_{\mathbf{h}\geq 0} \ \lambda\|\mathbf{h}\|_1 + \frac{\rho_k}{2}\|\mathbf{u}^{k+1} - \mathbf{h} + \frac{\mathbf{y}_2^k}{\rho_k}\|_2^2 \tag{15}$$

The (15) can be efficiently solved by shrinkage [22]. And a closed-form solution for the i-th element of variable $\mathbf{h}$ is derived as:

$$h_i^{k+1} = \max(\mathrm{S}_{\frac{\lambda}{\rho_k}}[u_i^{k+1} + \frac{(y_2^k)_i}{\rho_k}], 0) \tag{16}$$

where u_i^{k+1} is the i-th element in vector $\mathbf{u}^{k+1}$, $(y_2^k)_i$ is the i-th element in lagrange multiplier vector $\mathbf{y}_2^k$. And $\mathrm{S}_\epsilon[m]$ is the shrinkage operator which is defined as:

$$\mathrm{S}_\epsilon[m] = \begin{cases} m - \epsilon & \text{if } m > \epsilon \\ m + \epsilon & \text{if } m < -\epsilon \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Algorithm 1 Optimizing scheme for the NSSP

```
Input: audio event signal $\mathbf{X} \in \mathbb{R}^{D\times T}$, sparsity parameter $\lambda$.
Output: an optimal solution set $(\mathbf{u}^*, \mathbf{z}^*, \mathbf{h}^*)$.
Initialize: $\mathbf{u}_0 = \mathbf{z}_0 = \mathbf{h}_0 = \mathbf{y}_1^0 = \mathbf{y}_2^0 = \mathbf{0}, \epsilon1 = 10^{-4}, \epsilon2 = 10^{-1}, c_0 = 1.9, \rho_0 = \min(D,T) \times \epsilon2, \rho_{\max} = 10^{10}, \eta = 1.02 \times \|\mathbf{X}\|_F^2, k = 0.$
While $\|(\mathbf{u}^k)^{\mathrm{T}}\mathbf{X} - \mathbf{z}^k\|_2/\|\mathbf{X}\|_{\mathrm{F}} \geq \epsilon1$ or $\|\mathbf{u}^k - \mathbf{h}^k\|_2 \geq \epsilon2$ do
    Update variable $\mathbf{u}^{k+1}$ as (12) given $\mathbf{u}^k$ $\mathbf{z}^k$, $\mathbf{h}^k$, $\mathbf{y}_1^k$ and $\mathbf{y}_2^k$
    Update variable $\mathbf{z}^{k+1}$ as (14) given $\mathbf{u}^{k+1}$, $\mathbf{h}^k$, $\mathbf{y}_1^k$ and $\mathbf{y}_2^k$
    Update variable $\mathbf{h}^{k+1}$ as (16), (17) given $\mathbf{u}^{k+1}$, $\mathbf{z}^{k+1}$, $\mathbf{y}_1^k$ and $\mathbf{y}_2^k$
    Update Lagrange multipliers as follows:
        $\mathbf{y}_1^{k+1} = \mathbf{y}_1^k + \rho_k((\mathbf{u}^{k+1})^{\mathrm{T}}\mathbf{X} - \mathbf{z}^{k+1})$
        $\mathbf{y}_2^{k+1} = \mathbf{y}_2^k + \rho_k(\mathbf{u}^{k+1} - \mathbf{h}^{k+1})$
    Update penalty parameter as follows:
        $\rho_{k+1} = \min(\rho_{\max}, C\rho_k)$, where
        $C = \begin{cases} c_0 & \text{if } \rho_{max} * \max(\sqrt{\eta}\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2, \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2, \|\mathbf{h}^{k+1} - \mathbf{h}^k\|_2) \,/\, \|\mathbf{X}\|_{\mathrm{F}} < \epsilon2 \\ 1 & \text{otherwise.} \end{cases}$
    Update $k : k \leftarrow k + 1$.
End while
```

Meanwhile, it is also necessary to get Lagrange multipliers $\mathbf{y}_1, \mathbf{y}_2$ and penalty parameter ρ updated. The scheme for optimizing NSSP is shown in Algorithm **1**. And each output $\mathbf{u}^*$ of Algorithm **1** is gathered as the input of multiple classifiers. In addition, to fairly compared with the SP, the dimension of $\mathbf{u}$ in the NSSP is also set to one.

3. Performance Evaluation

3.1. Database description and Experimental setup

The AudioEvent database [23] is selected for evaluation. All audio event signals in this database are converted to 16 kHz sampling rate, 16 bits/sample and mono channel. There are 5043 samples in the public part of the database. The samples are in various length and belong to 28 different kinds of audio events. Similar to [24], the database is randomly split into training set (75%) and test set (25%).

To setup the experiments, there are mainly five steps: 1) *frame-level features extraction*, 39-dimensional MFCCs, log-energy and their delta and delta-delta extracted by a 25 ms window and a 10 ms shift size, are adopted; 2) *segment-level features extraction*, the BoAW is used to describe audio segments. And to extract the BoAW, K-means [25] clustering is applied to generate a codebook with 2000 centers and the input patch size for the BoAW is 40; 3) *segment-level features enrichment*, the TVM smooth, χ_2 non-linear mapping and $l2$ normalization [26] are utilized. More specifically, the expended coefficients in χ_2 kernel is set to 1 and homogeneity degree is 0.5; 4) *temporal features extraction*, this step is realized by the proposed SP and NSSP whose parameters are shown in Algorithm 1; 5) *classification*, considering the size of our database and the robustness, Support Vector Machine (SVM) [27] with three widely used kernels: χ_2, Radius Bias

Function (RBF) and linear kernel are employed for recognition. The parameters of χ_2 are set as aforementioned and the bandwidth of the RBF is set as default in Libsvm toolbox [28]. And other parameters (e.g., the penalty parameter C) are selected by 4-folds cross-validation.

3.2. Experimental Results and Discussions

Experiments are conducted to compare the proposed methods with other widely used methods in AER. For a fair comparison, some currently pooling methods are chosen to demonstrate the effectiveness of our methods. Furthermore, since DNN-based methods has been proven significant advancements in several tasks including AER, it is also necessary to compare our methods with these DNN-based modern methods.

3.2.1. Comparison with other pooling methods

Since our methods aim at pooling multiple segments of each entire signal into a single vector of temporal features, for a fair comparison, the selected typical pooling methods should also do such pooling. To this end, the BoAW and Rank Pooling (RP) [15] are chosen as baselines. More specifically, one baseline named BoAW$_{all}$ is designed to extract BoAW features for each entire signal, which always discards the temporal features between segments. The other baseline, i.e., the RP-based method, is a pooling method which can preserve partial temporal features between segments and shows good performance on the AudioEvent database [15]. And the SVM with three kernels (SVM$_{\chi_2}$, SVM$_{lin}$, SVM$_{rbf}$ in Table 1) is applied for AER. Furthermore, to make a global measurement, the mean value (Mean in Table 1) and standard deviation (Std in Table 1) of every three accuracies on different kernels are also calculated. In addition, as shown in section 2.2 and 2.3, it is not hard to find that the sparsity parameter λ may affect the performance of the NSSP. Thus, we further conduct experiments to investigate the influence of λ on the NSSP. The experimental results for the BoAW$_{all}$, RP, SP and NSSP with different sparsity parameter λ are shown in Table 1.

Table 1: *Performance comparison of four pooling methods*

Methods based on	Param λ	SVM$_{\chi_2}$	SVM$_{lin}$	SVM$_{rbf}$	Mean	Std
		Accuracy(%) on Classifiers				
BoAW$_{all}$	–	76.57	67.16	74.34	72.69	4.92
RP	–	78.38	71.86	67.82	72.69	5.33
SP	–	80.03	79.37	73.02	**77.47**	**3.87**
NSSP	1	72.77	73.43	73.01	73.07	0.33
	10^{-1}	80.12	79.79	80.86	80.26	0.55
	10^{-2}	80.36	80.12	80.36	**80.28**	**0.14**
	10^{-3}	78.38	80.03	81.27	79.89	1.45
	10^{-4}	78.55	80.20	79.95	79.57	0.89
	10^{-5}	78.05	80.03	79.95	79.34	1.12
	10^{-16}	78.03	80.03	79.95	79.34	1.13

From Table 1, it is easy to find that:

1) Comparing with the BoAW$_{all}$, the SP and NSSP achieve higher performance and more robustness. In detail, our methods bring 4.78% and 7.59% mean accuracy improvements over the BoAW$_{all}$ and have much lower standard deviations (i.e., 3.87 and 0.14, respectively) than that of BoAW$_{all}$ (i.e., 4.92). These advancements can be attributed to the positive influence of the effective temporal features.

2) Comparing with the RP, the SP and NSSP show their advances on both mean accuracy and standard deviation. Specifically, the mean accuracies of our methods are absolutely promoted by 4.78% and 7.59% beyond the RP, respectively. Besides, the standard deviations of the SP and NSSP are much lower than that of the RP (i.e., 5.33). The above advances indicate that for effectively extracting temporal features, our methods are more suitable and robust methods than the RP. These advancements are caused by jointly considering the signal reconstruction ability and temporal dynamics which is important for discrimination and robustness of AER

3) In Table 1, the proposed NSSP achieves the best mean accuracy (i.e., 80.28%) and highest performance (i.e., 81.27% when classified by the SVM$_{rbf}$). Moreover, it shows extremely robust to all three classifiers with the best standard derivative equals to 0.14. Also, comparing with the SP, the NSSP further absolutely boosts the mean accuracy by 2.81%. This confirms that the two constraints introduced by NSSP are reasonable and meaningful for AER.

4) The sparsity parameter λ has substantial effect on the NSSP. When λ is set to a relative large value (e.g., λ=1), it leads to worse performance. It is mainly caused by discarding too many dimensions to remain sufficient useful information. In addition, there is a trend that the mean accuracy first reaches a peak then gradually decreases when λ in a suitable range (e.g., λ from 10^{-1} to 10^{-5}). Besides, when λ is set to almost zero (e.g., λ=10^{-16}), it also provides similar performance to the performance achieved by setting λ equals to 10^{-5}. This similarity can be attributed to the inherent sparsity of the BoAW. Moreover, it also proves that the non-negative constraint is quite necessary for the NSSP.

3.2.2. Comparison with modern methods

Our proposed methods are also compared with CNN- and Convolutional LSTM (ConvLSTM)-based methods which both achieve fairly good performances in AER [13, 23]. The experimental results of these comparisons are shown in Table 2. We can find that the SP brings 2.13% and 1.09% absolute improvements compared with CNN- and convLSTM-based methods, respectively. Furthermore, the NSSP further boosts the accuracy over the SP by 1.24%. These comparisons also demonstrate the good performance of our methods.

Table 2: *Performance comparison with modern methods*

Methods based on	Accuracy (%)
CNN	77.90
ConvLSTM	78.94
SP	**80.03**
NSSP	**81.27**

4. Conclusion

This paper provides a new pooling strategy to effectively extract the temporal features between segments and proposes two efficient pooling methods for AER. The above strategy focus on considering both temporal dynamics and the signal reconstruction ability of the temporal features by leaning a low-dimensional subspace. And the basis of this subspace is used as an efficient representation for each signal. Moreover, a close-form solution and an alternating direction optimization scheme are also proposed for our proposed methods respectively. Multiple experiments on the AudioEvent database demonstrate the effectiveness of our methods.

5. Acknowledgements

This research is supported by the National Natural Science Foundation of China under grant No. U1736210 and 61471145.

6. References

[1] S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, and J. Hu, "An ensemble stacked convolutional neural network model for environmental event sound recognition," *Applied Sciences*, vol. 8, no. 7, pp. 1152–1172, 2018.

[2] C. Mydlarz, J. Salamon, and J. P. Bello, "The implementation of low-cost urban acoustic monitoring devices," *Applied Acoustics*, vol. 117, pp. 207–218, 2017.

[3] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, "Polyphonic sound event detection by using capsule neural networks," *IEEE Journal of Selected Topics in Signal Processing*, 2019.

[4] A. Kumar and B. Raj, "Features and kernels for audio event recognition," *arXiv preprint arXiv:1607.05765*, 2016.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[6] B. Elizalde, A. Shah, S. Dalmia, M. H. Lee, R. Badlani, A. Kumar, B. Raj, and I. Lane, "An approach for self-training audio event detectors using web data," in *25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1863–1867.

[7] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Conference of the International Speech Communication Association (INTERSPEECH)*. IEEE, 2012, pp. 2105–2108.

[8] A. Plinge, R. Grzeszick, and G. A. Fink, "A bag-of-features approach to acoustic event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3704–3708.

[9] H. Lim, M. J. Kim, and H. Kim, "Robust sound event classification using lbp-hog based bag-of-audio-words feature representation," in *Conference of the International Speech Communication Association (INTERSPEECH)*. IEEE, 2015, pp. 3325–3329.

[10] T. Komatsu and R. Kondo, "Detection of anomaly acoustic scenes based on a temporal dissimilarity model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 376–380.

[11] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," *arXiv preprint arXiv:1706.02293*, 2017.

[12] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[13] M. Meyer, J. Beutel, and L. Thiele, "Unsupervised feature learning for audio analysis," *arXiv preprint arXiv:1712.03835*, 2017.

[14] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *arXiv preprint arXiv:1604.00861*, 2016.

[15] L. Zhang, J. Han, and S. Deng, "Unsupervised temporal feature learning based on sparse coding embedded boaw for acoustic event recognition," in *Conference of the International Speech Communication Association (INTERSPEECH)*. IEEE, 2018, pp. 3284–3288.

[16] A. Cherian, B. Fernando, M. Harandi, and S. Gould, "Generalized rank pooling for activity recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2017, pp. 3222–3231.

[17] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan (Part I: Communications)*, vol. 67, no. 5, pp. 19–27, 1984.

[18] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[19] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[20] N. Shinozaki, M. Sibuya, and K. Tanabe, "Numerical algorithms for the moore-penrose inverse of a matrix: direct methods," *Annals of the Institute of Statistical Mathematics*, vol. 24, no. 1, pp. 193–203, 1972.

[21] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in neural information processing systems (NIPS)*, 2011, pp. 612–620.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[23] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Conference of the International Speech Communication Association (INTERSPEECH)*. IEEE, 2016, pp. 2982–2986.

[24] S. Deng, J. Han, C. Zhang, T. Zheng, and G. Zheng, "Robust minimum statistics project coefficients feature for acoustic environment recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 8232–8236.

[25] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[26] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence (PARMI)*, vol. 39, no. 4, pp. 773–787, 2017.

[27] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[28] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.