DISCRIMINATIVE SPARSE-BASED FEATURE EXTRACTION AND DICTIONARY LEARNING FOR SOUND CLASSIFICATION APPLICATIONS

Sanaz Seyedin\(^1\), Ramin Pichevar\(^2\), Jean Rouat\(^1\)

\(^1\) NECOTIS Research Group, Département de génie électrique et de génie informatique, Université de Sherbrooke, Sherbrooke, QC, Canada.
\(^2\) Communication Research Center, Ottawa, ON, Canada.

s.seyedin@usherbrooke.ca, ramin.pichevar@usherbrooke.ca, jean.rouat@usherbrooke.ca

www.gel.usherbrooke.ca/necotis

ABSTRACT

This paper presents a novel sparse-based classification algorithm for audio applications such as sound classification. We propose performing the sparse feature extraction, the dictionary learning, and classification processes simultaneously. This discriminative learning procedure for adapting the dictionaries and classifier to each specified audio task, instead of employing some pre-defined dictionaries is the main novelty of our work. According to our experiments, applying this algorithm on some Mel-scale spectral features, such as MFCC (Mel Frequency Cepstral Coefficient), instead of raw temporal data can improve the accuracy and execution time significantly. Our proposed discriminative MFCC-sparse features when evaluated on real data consisting of five audio classes, substantially outperformed the non-discriminative ones. The lengths of test segments in our method are less than 0.5 second. This potential of usage for real-time applications is another advantage of our proposed approach.

1. INTRODUCTION

Audio classification aims to categorize various sounds into different classes of percussion, harmonic signal, noise, etc., or go in more detail and specify the exact classes as speech, music, environmental sounds, animal vocals, etc. Audio classification is useful for object recognition in some robotic applications such as robot navigation, and also in speech recognition. It can also be used in other multimedia applications such as audio retrieval. There are different methods in the literature [1-6] for audio classification. Most of these methods comprise two steps that should be done separately or in parallel:

1. Feature extraction: is the most critical part of the algorithm, and choosing appropriate features will lead to better results. Sparse representation may be a suitable and interesting strategy in this step [3,4,7].

2. Audio classification, based on the extracted features: Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), and Support Vector Machines (SVM) are some examples among state of the art methods.

In this work, we are interested in classifying sounds into specified categories. It is advisable that the classification be simple and if possible linear for potential embedded hardware systems. For other practical reasons, it is desirable that the system be adaptable to new environments and sounds. Therefore, we study an original approach that includes all of these aspects: i) linear classification for a simple recognition process, ii) discriminant learning in which features and classifiers are found during training, so that features are found by taking into account the constraints of linear classification and iii) sparse representation as the appropriate features for sound processing. In fact, sparse features can solve the problem of dimension reduction efficiently [8], and have greater robustness in the presence of noise [7,9].

In order to linearly classify features obtained from various sounds, the input vectors should be mapped to a very high-dimension feature space [10]. Finding this new higher dimensional space is intractable if constraints are not taken into account. Discriminant analysis [11] can be used to find that transformation based on the separation quality in the new space. Working in this new high dimensional space will be difficult if most of the sound feature vector components are non-zero. Adding another constraint namely sparsity in sparse representation techniques may be an appropriate solution [7,8].

In this paper, we propose using a novel method based on sparse representation for sound classification. To this end, we take into account all three above-mentioned aspects, i.e. linear classification, discriminant learning, and using appropriate sparse representation. Therefore, we extract sparse features, perform the dictionary learning and classification process simultaneously, similar to the approach used in [12] for image classification. We adapt this algorithm to make it efficient for audio signals. We perform training and classification with both discriminant and non-discriminant sparse-based approaches and prove that adding the discriminant term can substantially improve the classification accuracy. We also employ our suggested algorithm in two domains: temporal and spectral. We propose performing this algorithm on extracted
spectral features from audio signals with much lower dimension than raw temporal data, to improve the accuracy and execution time. In comparison to the state of the art in sound classification, the most important novelty of the proposed algorithm is the discriminative learning procedure to adapt the dictionaries to the specified task, instead of employing pre-defined dictionaries, and also find the linear classifier parameters in the same procedure. This makes our technique different from other existing audio classification methods using sparse coding, such as [3,4]. To our best knowledge, this discriminative learning was not used before for sound classification tasks. This suggested procedure can be used in different audio processing applications such as sound classification, where an exact reconstruction of the signals is not required. In addition, we could achieve good classification accuracies over short segments with lengths of less than 0.5s.

The paper is organized as follows. In Section 2, we present a brief review of the previous work in this field. Section 3, describes our proposed sparse-based technique for extracting the features, dictionary learning, and sound classification. The experimental results are presented in Section 4. Finally, discussion and conclusion are given in Sections 5 and 6, respectively.

2. PREVIOUS WORK

The approach used in [1] is one of the first and basic ones used for audio content analysis. The authors used basic features like amplitude, fundamental frequency, and pitch for this purpose. Therefore, the results on limited number of classes are not satisfactory enough and the algorithm needs to be improved. In [2], a simple feature extraction method is introduced based on different time and frequency features for sound classification. The acoustical features in [2] include loudness, pitch, brightness, bandwidth, and harmonicity. Finally, the feature vector consists of the duration of the sounds plus the mean, variance, and autocorrelation of the mentioned features. For classification, a simple model is trained using the mean vector and covariance matrix of the feature vectors on training data. In the testing mode, a distance measure is calculated for each new test sound with the defined model. The database includes various sounds of animals, machines, musical instruments, speech, and nature. A time-frequency representation has been suggested in [3] for classifying environmental sounds. The authors in [3] proposed an approach based on Matching Pursuit (MP) to obtain time-frequency features. They use 256-point segments and decompose each temporal segment using MP with a dictionary of temporal hand-made Gabor atoms. The features include the mean and standard deviations of frequency and scale parameters of the obtained atoms. Finally, the classification decision is performed using GMMs or K-Nearest Neighbor (kNN). Their proposed features outperform Mel Frequency Cepstral Coefficients (MFCC) for some classes, while it has much less accuracy for other sounds. For this reason, they have suggested concatenating their proposed coefficients with MFCC. However, this increases the dimension of the resulting feature vector. The classification accuracies are reported for 4-s segments in [3]. This method had a similar idea with [4] where Gaussian bases were used for classifying different music sounds, which include rock, classical, country, jazz, folk, and pop. In the end, classification was carried out by a linear discriminant analysis (LDA)-based algorithm. A hierarchical classification has been suggested for classifying different audio signals in [5]. In this method, speech is separated from non-speech segments at the first step, and then the second audio classification is performed on non-speech parts to separate them into music, environmental sounds, silence, etc. This hierarchical classification makes the process more complex. A musical instrument classification for classifying polyphonic instruments is suggested in [6]. This method is useful only for determining the instruments used in the music signal and successful only if at least one isolated partial exists for each instrument somewhere in the signal. The partials are found based on formant frequencies. The features are then extracted and the instrument which has a larger value based on a threshold of the features wins.

While these methods can be efficient in specific situations, we expect our proposed approach to have a stronger potential because of the combination of a simple classifier with sparse features and the use of discriminative learning. In other words, both the dictionary and classifier parameters are found in the training stage, contrary to other methods. Thus the classification can be performed very fast by finding the sparse features for each test segment. This approach facilitates its usage for real-time applications.

3. SPARSE-BASED FEATURE EXTRACTION AND DICTIONARY LEARNING FOR CLASSIFICATION

In sparse coding, the signal $X$ can be decomposed into the overcomplete kernels or dictionaries, $D$, with $K$ atoms as follows [7]

$$ X = DA, $$

where $X \in \mathbb{R}^{n \times M}$, $D \in \mathbb{R}^{n \times K}$, and $A \in \mathbb{R}^{K \times M}$ is called the sparse codes. In addition, $n$ and $M$ are the dimension and number of sets of the input signal, respectively. The dictionary $D$ and sparse codes $A$ are found by solving the following optimization problem [7,12]

$$ < D, A > = \arg \min_{D,A} \|X - DA\|^2_2 \text{ s.t. } \forall i, \|a_i\|_0 \leq T_0 $$

where $T_0$ is the sparsity constraint factor, and $a_i$s are column vectors of size $K$ of $A$.

The dictionary can be chosen based on some a priori knowledge about the signal [3,7], or learned using training data [12]. We prefer the latter case since the dictionary can be adapted to the input signals in the training procedure. In some sound processing applications like sound classification, the exact reconstruction of the signals is not required and can deteriorate the classification performance.
This is due to the fact that for classification, the purpose is finding sparse features which makes the task more accurate while giving less priority to perfect reconstruction of the original signal. For this reason, we use a novel method to jointly learn the dictionary, sparse features, and the classifier. Figure 1 shows the proposed approach. A label consistent K-SVD (LC-KSVD) is suggested for classifying images in [12] in line with the above-mentioned idea. In [12], each atom of the dictionary is assigned to only one class and is kept unchanged till the end of the algorithm. Therefore, it is called label-consistent K-SVD. In this approach, a cost function composed of three parts is used: the cost function used regularly for sparse coding (Eq. (2)), a function showing the discriminability of the sparse codes, and a term representing the classification error [12]:

\[
< D, C, T, A > = \arg \min_{D,C,T,A} \| X - DA \|_2^2 + \alpha \| Q - TA \|_2^2 + \beta \| L - CA \|_2^2 \quad \text{s.t.} \forall i, \| a_i \|_0 \leq T_0
\]  

where \( Q = [q_1 \ldots q_M] \in \mathbb{R}^{K \times M} \) is the discriminative sparse codes of input signals \( X \) for classification. It consists of zero and one values where the non-zero values belong to those indices for which the input signal \( x_i \) and the dictionary item \( d_k \) share the same label. \( L = [l_1 \ldots l_M] \) is the class labels of input signals \( X \). This matrix includes 0 and 1 values where the non-zero ones indicates the class of the corresponding input signal. \( T \) is a linear transformation matrix relative to sparse codes, and \( C \) represents the linear classifier parameters. \( \alpha \) and \( \beta \) are constants which control the relative contribution between the corresponding terms. Equation (3) can be rewritten as follows [12]:

\[
< D, C, T, A > = \arg \min_{D,C,T,A} \| X - \frac{D}{\sqrt{\alpha Q}} \left( \frac{\sqrt{\alpha Q}}{\sqrt{\beta L}} \right) - \frac{D}{\sqrt{\alpha T}} A \|_2^2 \quad \text{s.t.} \forall i, \| a_i \|_0 \leq T_0
\]

Therefore, the optimization can be done simultaneously by using the K-SVD (K-Singular Value Decomposition) algorithm [13] by defining \( X_{\text{new}} \) and \( D_{\text{new}} \) as follows:

\[
X_{\text{new}} = \left( \frac{X}{\sqrt{\alpha Q'}} \right), D_{\text{new}} = \left( \frac{D}{\sqrt{\alpha T'}} \right)
\]

This makes the optimization problem to be solved faster. K-SVD is the generalization of K-means clustering algorithm, i.e. each input signal is represented by a linear combination of codewords (atoms) [13]. Furthermore, each dictionary atom and its relevant sparse coefficients are updated simultaneously in this algorithm to accelerate the convergence [13].

The algorithm needs the initialization of the dictionaries to converge faster and give satisfactory results. For this reason, we first obtain the dictionaries for each class using the original KSVD algorithm [13] separately. We then find the large dictionary for LC-KSVD algorithm by concatenating different dictionaries found for each class. Next, we initialize the sparse codes, \( X, \) the transformation matrix, \( T, \) and the classifier model, \( C, \) one after another. Furthermore, we assign the class label values to \( Q \) and \( L \) matrices. Finally, we solve (4) using (5) and the original KSVD algorithm. Note that the desired values for the dictionary and classifier matrix will be obtained after a simple normalization process with respect to L2-norm of the calculated dictionary [12].

\[
\hat{D} = \left\{ \frac{d_1}{\|d_1\|_2}, \frac{d_2}{\|d_2\|_2}, \ldots, \frac{d_K}{\|d_K\|_2} \right\}
\]

\[
\hat{C} = \left\{ \frac{c_1}{\|d_1\|_2}, \frac{c_2}{\|d_2\|_2}, \ldots, \frac{c_K}{\|d_K\|_2} \right\}
\]

The purpose of this approach is to do the classification very easily using the sparse features and the linear classifier on each desired input:

\[
i = \arg \max_i (\hat{C}A)
\]

4. EXPERIMENTAL RESULTS

The presented algorithm has been tested on images [12], and we extend and modify it to make it efficient for sound classification. We evaluated the mentioned algorithm on both synthetic data and real data. The synthetic data we prepared for testing and training the algorithm consist of two classes:

1. Sinusoidal harmonics with different frequencies, and sampling frequency of \( F_s = 8\) KHz, to simulate harmonic signals and models of percussions with sinusoids.
2. Some data having Gaussian distribution with different means and standard deviations to simulate noise-like data. We refer to these data as noise.

The training procedure needs a large data set to converge well. Therefore, we simply replicated our synthetic training data several times to satisfy this criteria. This folding procedure is similar to assigning some weights to data points in some classes, and performing the training step more carefully for that specified class which needs more training. Therefore, the atoms are fitted more precisely. Consequently, the classification accuracy is improved. Then we ran our proposed algorithm mentioned in Section 3 on time-domain signals with a dictionary size of $K = 118$, i.e. $K_1 = 80$ and $K_2 = 38$ as the dictionary sizes for each class. Some of the obtained atoms found for sinusoidal harmonics are shown in Figure 2. This figure also illustrates that the distribution of the Gaussian noise class fits well with a Gaussian function, as expected. The atoms we obtained for synthetic data can also reflect the characteristics of the atoms for some real harmonic signals, such as percussions, speech and music, or noise-like ones. The classification test consists of two stages: i) on the training data itself, ii) on test data, which are different from training data. We were able to achieve the classification accuracy of 100% for training set with this algorithm. The average classification accuracy is more than 60% over the test set with the limited data we used.

We also evaluated the capability of the sparse-based features and the proposed approach for sound classification on some real data consisting of five classes of speech, music, inside car, factory, and restaurant. All files have a sampling rate of $F_s = 8$ KHz. We collected these data from different sources:

- speech data: we extracted some files from the clean training (for training) and Test set A utterances (for testing) of Aurora 2 task [14]. These files are carefully chosen such that they are long enough and appropriate for sound classification.
- music data: we created a small database from individual music CDs.
- inside car, factory, and restaurant: we gathered them from some sample noise signals in NOISEX-92 [15].

We employed 20 minutes of speech data, 11 minutes of music, and around 9 minutes for each of inside car, factory, and restaurant classes. We used 70% of each class for training and the rest 30% for testing. We also observed that increasing the data set improves the classification accuracies.

We propose two different scenarios:

1. Implementing the proposed algorithm in Section 3 (Eq. (4)) on temporal (time-domain) features. To this end, we take non-overlapping frames of length 1s (8000 samples) as the input signal $X$.

2. Proposing a new feature extraction procedure by finding the sparse-based features (the algorithm mentioned in Section 3, Eq. (4)) on Mel Frequency Cepstral Coefficients (MFCC) [16]. In other words, input signal $X$ in Eq. (4) is replaced by MFCCs.

The classification results showed that applying our proposed sparse-based algorithm on MFCC features, rather than temporal ones, could lead to better classification results. We calculate MFCC features with the same procedure as in [16]. Thus, we take frame lengths of 25 ms with frame-shifts of 10 ms from the described data set. We apply these frames to a pre-emphasis filter ($H(z) = 1 - 0.97z^{-1}$) and weight the result with a Hamming window. Then, we pass the power spectrum of the resulting output frame through the Mel-filter-bank which consists of 23 triangular filters. Finally, we compute 12 cepstral features by applying IDCT (Inverse Discrete Cosine Transform) to the logarithm of the resulting values. It is worth mentioning that because of the limitations in finding the power spectrum, we cannot use long frames without any overlaps in this strategy. For that reason, we concatenate the features of 40 frames to be able to work with the information in longer segments. Needless to mention that the algorithm over MFCC coefficients is much faster than the temporal ones, because we have only 480 features for each segment, instead of 8000 samples. In addition, we can obtain better classification results with even shorter segments (less than 0.5s). This is the most advantageous point about the proposed sparse-based MFCC features.

We evaluated the proposed algorithm on five different audio classes on both temporal and MFCC features. To this end, we jointly learn the dictionary atoms and the classifier model in the training phase, and find the sparse codes and the class indices in the testing phase according to Figure 1. We set the dictionary size to $K = 1300$, i.e. $K_1 = 350$, $K_2 = 300$, $K_3 = 200$, $K_4 = 200$, and $K_5 = 250$ for time-domain case. The dictionary size for MFCC case is $K = 1320$, i.e., $K_1 = 380$, $K_2 = 270$, $K_3 = 220$, $K_4 = 220$, and $K_5 = 230$ as the dictionary sizes for each class. In addition, $\alpha$ and $\beta$ are fixed to 16 and 4 respectively. Moreover, we observed that using 20 sparse features is sufficient to obtain satisfactory results ($T_0 = 20$). We have tuned these values experimentally by evaluating the classification accuracies on the test set with different values. Figure 3 compares the classification accuracies obtained using two proposed scenarios. In order to show the advantage of discriminative learning approach in this task, we also tested the classification ability of the simple sparse-based approach using (2) with the same dictionary sizes we chose for MFCC-sparse features. In other words, we learn the dictionaries for each class separately using the standard sparse coding strategy. Then we find the sparse codes for all classes separately for each test signal. Finally, the class which minimizes the standard sparse cost function in (2) is chosen as the desired one. The results obtained by implementing this algorithm on MFCC features is also shown in Figure 3, namely MFCC-sparse1, compared to MFCC-sparse2 for discriminative case. Note that we have tested the algorithm with different dictionary sizes, and have reported those which give the best results. The confusion matrix for our proposed discriminative learning approach, MFCC-sparse features
Fig. 2. Some of the properties of the atoms obtained by implementing the proposed sparse-based procedure on synthetic data.

according to (4), is also shown in Table 1. The rows of this matrix denote the classes we want to classify, while the columns show the classified ones.

5. DISCUSSION

The results in Figure (3) clearly show the success of our proposed algorithm for sound classification. According to this figure, MFCC sparse-based features are more appropriate features for sound classification than temporal ones. MFCC-sparse features outperform temporal-sparse ones regardless of the fact that we used much longer segments in the latter case to improve the results. MFCC features can accurately reflect the important characteristics of the signals required for classification. In other words, the redundant information which is not useful for classification has been omitted in MFCC features. In other words, the discriminative dictionary atoms can be fitted more precisely while using MFCC features with much lower dimension than time-domain signal. According to Figure 3, our proposed method can successfully classify speech, music and car signals. The reason why the classification accuracy is lower for factory and restaurant audio signals is that there are some overlaps between them and speech as well as music signals. For instance, there are some people speaking in the restaurant while the files were recorded. The confusion matrix in Table 1 also shows the same concept. The discriminative learning procedure we employed in our method is the main reason of the accurate classifications we could achieve. The results in Figure 3, also prove this claim, because our proposed MFCC-sparse2 features outperform MFCC-sparse1 significantly.

6. CONCLUSION

In this paper, we presented a novel sparse-based classification algorithm which can be used in some audio applications such as sound classification. The sparse feature extraction, the dictionary learning and classification are done simultaneously in our proposed approach. We observed that applying this algorithm on Mel-scale spectral features, such as MFCC, instead of raw temporal data can improve the accuracy and execution time significantly. We can interpret these features as MFCC-sparse coefficients in contrast to temporal-sparse ones reported in previous audio classification strategies. The discriminative learning procedure we are using for adapting the dictionaries and classifier to our specified audio task, instead of employing some pre-defined dictionaries is another important novelty of this work. This approach substantially improves the classification accuracies compared to non-discriminative case. The experimental results showed that our proposed method is successful for classifying different harmonic signals (speech and music) from noise-like ones (car, factory, restaurant). Although these signals have some common parts and overlaps, our suggested approach can successfully discriminate between them in very short segments (shorter than 500ms). It is worth mentioning that this advantage makes our algorithm useful for real-time applications, while many other audio classification tasks were tested with longer segments (much longer than 1s).

We intend to improve the results by using a larger database in our future work.

7. ACKNOWLEDGMENT

We would like to thank Ron Rubinstein for providing free source code of original KSVD algorithm [17]: (http://www.cs.technion.ac.il/~ronrubin/software.html).

8. REFERENCES

Table 1. Confusion matrix for 5-class sound classification using our proposed MFCC sparse-based method according to (4).

<table>
<thead>
<tr>
<th>Classes</th>
<th>Speech</th>
<th>Music</th>
<th>Car</th>
<th>Factory</th>
<th>Restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>96.66</td>
<td>2.16</td>
<td>0</td>
<td>0.43</td>
<td>0.75</td>
</tr>
<tr>
<td>Music</td>
<td>0</td>
<td>99.66</td>
<td>0.34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Factory</td>
<td>13.07</td>
<td>0</td>
<td>0</td>
<td>79.55</td>
<td>7.39</td>
</tr>
<tr>
<td>Restaurant</td>
<td>18.22</td>
<td>4.67</td>
<td>0</td>
<td>0.93</td>
<td>76.17</td>
</tr>
</tbody>
</table>

Fig. 3. The classification accuracy(%) of the proposed approach for 5 audio classes in two scenarios of temporal (temporal-sparse) and MFCC features (MFCC-sparse2). MFCC-sparse1 is the sparse features obtained by (2) (without using the discriminative term).


