Digital Signal Processing 23 (2013) 960–970

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Dictionary learning based sparse coefficients for audio classification with max and average pooling

Syed Zubair *, Fei Yan, Wenwu Wang

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK

ARTICLE INFO

Article history:
Available online 26 January 2013

Keywords:
Audio classification
Sparse coefficients
Dictionary learning
Support vector machines

ABSTRACT

Audio classification is an important problem in signal processing and pattern recognition with potential applications in audio retrieval, documentation and scene analysis. Common to general signal classification systems, it involves both training and classification (or testing) stages. The performance of an audio classification system, such as its complexity and classification accuracy, depends highly on the choice of the signal features and the classifiers. Several features have been widely exploited in existing methods, such as the mel-frequency cepstrum coefficients (MFCCs), line spectral frequencies (LSF) and short time energy (STM). In this paper, instead of using these well-established features, we explore the potential of sparse features, derived from the dictionary of signal atoms using sparse coding based on e.g. orthogonal matching pursuit (OMP), where the atoms are adapted directly from audio training data using the K-SVD dictionary learning algorithm. To reduce the computational complexity, we propose to perform pooling and sampling operations on the sparse coefficients. Such operations also help to maintain a unified dimension of the signal features, regardless of the various lengths of the training and testing signals. Using the popular support vector machine (SVM) as the classifier, we examine the performance of the proposed classification system for two binary classification problems, namely speech–music classification and male–female speech discrimination and a multi-class problem, speaker identification. The experimental results show that the sparse (max-pooled and average-pooled) coefficients perform better than the classical MFCCs features, in particular, for noisy audio data.

1. Introduction

Audio signals acquired from an uncontrolled natural environment have different types of contents e.g. speech, music and environmental sounds. For example, in radio broadcast system, a mixture of different types of sounds is usually encountered such as speech for news broadcasting, music for song broadcasts or a mixture of both. In content based retrieval system, different contents such as voiced, unvoiced speech and music are required to be distinguished from each other. Different encoders related to different types of contents are used. When broadcasting speech, only speech encoder should be activated while disabling encoders of all other content types. This helps to reduce the power consumption of the system without overloading it with simultaneous activation of other encoders and to reduce computational costs. Such a content based system requires the signal classification system for its front-end [1]. Audio classification is also useful for identifying the surrounding environments of a person, e.g., in a restaurant, near a sea-shore or in a shop [2]. Another example for the application of audio classification system is to find and track a specific audio document from an archive of piles of audio recordings. All of these exemplar applications require a powerful audio classification system.

Signal classification is in general a two-step process. First signal features are extracted from training data and then used to train a classifier. Second the trained classifier is used to discriminate the test signals based on their features. A lot of research in this area has been conducted in last two decades with the methods proposed mainly differing in the types of features and classification techniques used [3–5].

Various time, frequency and time–frequency representations have been used in the literature for generating audio features. For example, zero crossing rate (ZCR) [1,6] and short-time energy (STE) [7,8], together with their variations are the low level time domain features that have been used extensively. ZCR measures the change in algebraic signs of the signal amplitudes in a specified window. The contour waveforms of speech ZCR distribution show abrupt change in the amplitude as opposed to music contours. This difference of contours makes ZCR a discernible feature for speech and music discrimination. STE is another time domain feature that uses the signal energy to distinguish one type of signal from another. The frequency domain features that have been used include

* Corresponding author.
E-mail addresses: s.zubair@surrey.ac.uk (S. Zubair), f.yan@surrey.ac.uk (F. Yan), w.wang@surrey.ac.uk (W. Wang).

1051-2004/$ – see front matter
http://dx.doi.org/10.1016/j.dsp.2013.01.004

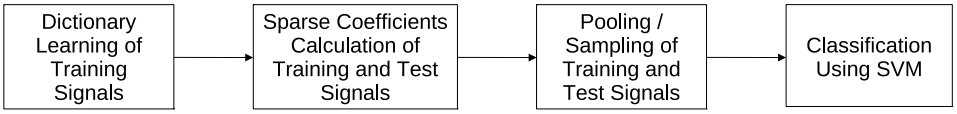


Fig. 1. Block diagram of the proposed audio signal classification system.

line spectral frequencies (LSF) [9], 4 Hz modulation energy, spectral centroid, spectral flux [10] and mel-frequency cepstral coefficients (MFCCs) [5,8]. As an audio signal has different frequency components, these features decompose the signal into its constituent frequency components/bands and use energy corresponding to each frequency band as a measure of discriminating feature. Some other features are based on psychoacoustic principles and human auditory systems [11], including perceptual loudness, roughness [12], and sharpness of the signal, as well as auditory filter-bank temporal envelops (AFTE) [11].

The performance of an audio classification system is dependent not only on the features used but also on the selection of an appropriate classifier. Hence the second stage in audio classification involves the selection of the type of classifiers. A number of different classifiers have been used for audio signal discrimination and classification including Gaussian mixture models (GMM) [1,13], K nearest neighbors (KNN) [10,9,14], neural network (NN) [15,16], hidden Markov model (HMM) [7,17] and support vector machine (SVM) [18] along with their variations.

In this paper, different from the majority of the existing methods, we propose to use a new type of features for audio classification, which is obtained by sparse coding of signals with a variety of pooling techniques. Sparse coding is an emerging technique in signal processing that aims to express a signal as a linear combination of a small number of signal components (also referred to as atoms or codewords) from a dictionary (i.e. the collection of the atoms).

Sparse representations have been successfully employed in many applications like denoising [19], coding [20] and source separation [21]. For signal encoding, sparse representation helps to reduce the encoding complexity of a signal and decrease the bandwidth requirement for its transmission. For denoising [19], basis vectors of the transformation matrix representing noise are different from those representing the actual signal, hence the coefficients showing the activity of noise basis vectors are different from those of actual signal. However less attention has been paid in the literature to their use for audio signal classification. The sparse coefficients have a high potential to be used as signal features for classification due to their discriminating property. Recently, [22] and [2] used sparse coefficients for drums and environmental sounds classification respectively. They named those coefficients MP-based features as matching pursuit (MP) algorithm was used to calculate them. Semi-supervised learning algorithms have also been used with sparsity constraints for audio classification in [23], a self-taught learning strategy in a semi-supervised fashion, whose complexity is of order $O(L^3)$ for L non-zero coefficients. The algorithm [24] employs deep belief networks for unsupervised learning of sparse coefficients. Dictionary learning based on high level audio features has been used for audio classification in a supervised fashion in [25].

Many signals are either naturally sparse, or they can be made sparse in some specific domain by using some predefined transforms such as the discrete Fourier transform (DFT) or the discrete cosine transform (DCT). Apart from using predefined transforms, learning transform matrix directly from training data has also been proposed recently [26,19,20]. This inherent or manufactured sparsity of audio signals will lead potentially to a lower computational complexity and less demand of the resources.

In this paper, based upon the discriminating properties of sparse coefficients, we propose an audio classification system where sparse coefficients are used as audio features with the application of the state of the art SVM classifier. The K-SVD dictionary learning algorithm [19] is used to learn a dictionary from training signals. The learnt dictionary is used to find sparse codes of training and test signals. The standard practice for training the classifier is to use training vectors of the same dimensions. Since in our case, different sizes of training and testing signals result in training and testing coefficient vectors with different dimensions. Hence we introduce novel max pooled and average pooled sparse coefficients for audio signal classification which not only solve the issue due to the mismatched dimensions but also select only those dictionary atoms that have a maximum or high contribution towards signal representations. They serve to summarize a coefficient matrix representing a signal to a vector that helps to drastically decrease computational complexity and memory requirements. Summarizing the matrix to a vector may lose some important signal information essential for discrimination. Hence we introduce sampled sparse as well as sampled mel-frequency coefficients (MFCCs) for the classification system. We evaluate the discriminating power of pooled and sampled sparse coefficients by comparing them with the sampled MFCCs particularly under noisy conditions.

This paper has been divided into the following sections. Section 2 discusses the whole audio classification system in detail including K-SVD algorithm for dictionary learning, orthogonal matching pursuit (OMP) for sparse coding and various pooling and sampling techniques. Section 3 presents the experiments performed together with the analysis of results. The conclusion is given in Section 4.

2. The proposed audio signal classification system

A block diagram of the proposed audio classification system is shown in Fig. 1. Dictionaries are learned from the training signals and used to find sparse coefficients of the training and the testing signals. Pooling/sampling is performed on those sparse coefficients to reduce the large amount of data to an appropriate level and to give the training vectors a unified length. More importantly, it is used to get the compact representation of features which are invariant to local transformations. These pooled/sampled coefficients are then used as features and fed to the SVM classifier [18] for audio classification task.

2.1. Dictionary learning of training signals

An important element in sparse coding is the design of an appropriate dictionary whose atoms are used to represent a signal sparsely. Dictionary is a transformation matrix that is used to represent a signal in a specific domain, e.g. the frequency domain, which can be obtained by a predefined function such as the DCT. Unlike other approaches [2] in which a predefined dictionary is used for sparse coding, we use a dictionary learning algorithm that adapts to the internal structure of the training signals under special constraints, e.g. sparsity.

The objective function for dictionary learning of an input signal $\mathbf{Y} \in R^{n\times m}$ with sparsity constraint is given as

$$\|\mathbf{Y} - \mathbf{AX}\|_F^2 \quad \text{s.t. } \forall q \ \|\mathbf{x}_q\|_0 \leqslant T_0 \tag{1}$$

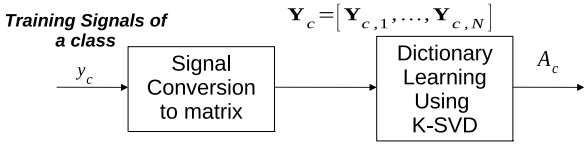


Fig. 2. Dictionary learning of training audio signals. Class specific time domain signals y_c are first converted to matrix $\mathbf{Y}_c$ and then used to learn a class specific dictionary $\mathbf{A}_c$.

where $\mathbf{A} \in R^{n\times l}$ is a dictionary matrix, $\mathbf{X} \in R^{l\times m}$ is a coefficient matrix with $\mathbf{x}_q$ being its q-th column vector, T_0 is a small positive value indicating the sparsity of vector $\mathbf{x}_q$ and $\|\cdot\|_0$ is the l_0 norm counting the number of non-zero values in vector $\mathbf{x}_q$. $\|\cdot\|_F$ denotes the Frobenius norm.

Dictionary learning is often achieved with a two-step iterative process. In the first step, given input signal $\mathbf{Y}$ and an initial dictionary matrix $\mathbf{A}$, sparse coefficient matrix $\mathbf{X}$ containing $\mathbf{x}_q$ vectors is calculated. In the second step, given the input signal matrix $\mathbf{Y}$ and coefficients matrix $\mathbf{X}$ calculated in the previous step, dictionary vectors (i.e. atoms) are updated. These two steps are iterated until the most appropriate dictionary matrix is found in the sense that a predefined cost function such as (1) is optimized.

An increased research interest in the area of dictionary learning has led to some state of the art algorithms like maximum likelihood (ML) based methods [27], method of optimal directions (MOD) [28], maximum a-posteriori probability (MAP) [26], K-SVD [19] and the majorization minimization (MM) [20] methods. As demonstrated in [20], the K-SVD algorithm produces more accurate dictionary than MOD and MAP, but offers comparable results to that of MM method. Moreover, it has better convergence properties and denoising capabilities [19]. Based upon these merits, we use the K-SVD algorithm for dictionary learning of our training signals.

2.1.1. K-SVD dictionary learning algorithm

In the two-step dictionary learning process, the K-SVD algorithm [19] uses (OMP) [29] for sparse coefficients calculation and singular value decomposition (SVD) for calculating and updating the dictionary atoms. In the dictionary learning step, $\mathbf{AX}$ is decomposed into K rank-1 matrices by selecting a dictionary element $\mathbf{a}_k$ and its corresponding coefficient vector $\mathbf{x}_T^k$ which is the k-th row in matrix $\mathbf{X}$, with its sparsity level denoted as T.

$$\|\mathbf{Y}-\mathbf{AX}\|_F^2 = \left\|\mathbf{Y}-\sum_{j=1}^{K}\mathbf{a}_j\mathbf{x}_T^j\right\|_F^2 = \left\|\left(\mathbf{Y}-\sum_{j\neq k}\mathbf{a}_j\mathbf{x}_T^j\right)-\mathbf{a}_k\mathbf{x}_T^k\right\|_F^2 = \left\|\mathbf{E}_k-\mathbf{a}_k\mathbf{x}_T^k\right\|_F^2 \quad (2)$$

where $\mathbf{E}_k$ is the error term formulated by excluding an arbitrarily selected dictionary element from $\mathbf{A}$. Now SVD is used to find the closest rank-1 matrix that effectively minimizes the error. After removing columns from $\mathbf{E}_k$ that do not use $\mathbf{a}_k$, the SVD of $\mathbf{E}_k$ yields $\mathbf{U}\Delta\mathbf{V}^T$, where the first column of $\mathbf{U}$ gives the updated dictionary atom $\mathbf{a}_k$ and the first column of $\mathbf{V}$ multiplied by $\Delta(1,1)$ gives the coefficient vector $\mathbf{x}_T^k$ corresponding to the dictionary atom. Iterating through the two steps of dictionary learning, the K-SVD produces a dictionary that fits the given signal $\mathbf{Y}$.

2.1.2. Learning dictionary atoms from training signals by K-SVD

Dictionary learning process for the set of training signals is shown in Fig. 2. For the purpose of classification, training signals $y_{c,i}$ of each class are passed through the K-SVD algorithm to get its corresponding dictionary, where c represents the class of the signal e.g. speech (sp), music (mus), male (m) and female (f), and i in $y_{c,i}$ represents the index of the training signal in that class of audio signals. Before applying K-SVD, the one-dimensional raw training signals $y_{c,i}$ are first decomposed into frames of equal length of size R^n without overlap and concatenated side by side to form a two-dimensional matrices $\mathbf{Y}_{c,i}$. This set of all two-dimensional signals belonging to one class are combined together to form one large matrix $\mathbf{Y}_c = [\mathbf{Y}_{c,1}, \mathbf{Y}_{c,2}, \ldots, \mathbf{Y}_{c,N}]$. This large matrix $\mathbf{Y}_c \in R^{n\times m}$ is fed to the K-SVD to get a dictionary $\mathbf{A}_c \in R^{n\times l}$ representing the dictionary of one class of the signals.

This dictionary $\mathbf{A}_c$ is used to obtain the sparse coefficients of both the training and testing signals for each class in the sparse coding stage.

2.2. Sparse coding

In this method, a natural signal is represented in terms of a small number of codewords or atoms taken from a dictionary either predefined or learned. Given a dictionary, many methods have been developed for finding the sparse coefficients to encode the signal such as matching pursuit (MP) [30], orthogonal matching pursuit (OMP) [29], basis pursuit (BP) [31], regression shrinkage and selection (LASSO) [32], focal under-determined system solver (FOCUSS) [33] and gradient pursuit (GP) [34]. Here we use OMP to find sparse coefficients which is the part of K-SVD two step dictionary learning process.

2.2.1. Orthogonal matching pursuit algorithm

To calculate sparse coefficients of an input signal with a given dictionary, the OMP algorithm [29] projects the input signal on the subspace spanned by the dictionary atoms. The atom which strongly correlates with the signal or its residual is selected and used for calculation of the coefficients. The whole algorithm works as follows:

- Initialize the residual $\mathbf{r}_0$ to be the input signal vector $\mathbf{y}_q$ and coefficient vector $\mathbf{x}_0$ to zero.
- At step k, a new atom is selected according to the following optimization problem

$$\lambda_k = \arg\max_{\omega\in\Omega}\left|\langle \mathbf{r}_{k-1}, \mathbf{a}_\omega\rangle\right| \quad (3)$$

  where $\langle\cdot,\cdot\rangle$ is a dot product, $|\cdot|$ is a modulus, Ω is the index set of all the atoms in the dictionary and λ_k is the index of the selected atom.
- Let $\Lambda_k = \{\lambda_1, \ldots, \lambda_k\}$ list the atoms that have been chosen at step k, then the k-th approximant (coefficient) is calculated as

$$\mathbf{x}_k = \arg\min_{\mathbf{x}}\|\mathbf{y}_q-\mathbf{x}\| \quad \text{s.t. } \mathbf{x}\in \text{span}\{\mathbf{a}_\lambda\colon \lambda\in\Lambda_k\} \quad (4)$$

  This minimization can be performed incrementally by the standard least squares techniques. The residual is then calculated as $\mathbf{r}_k = \mathbf{r}_{k-1} - \langle \mathbf{r}_{k-1}, \mathbf{a}_{\lambda_k}\rangle \mathbf{a}_{\lambda_k}$.

2.2.2. Sparse coding of training and testing signals

Sparse coefficients matrices of training signals $\mathbf{X}_{cd,i}$ from each class are obtained by using the OMP algorithm [29], here the d subscript represents the class specific dictionary that produces the coefficient matrix. The input training signals $\mathbf{Y}_{c,i}$ are projected on the subspace spanned by the dictionaries $\mathbf{A}_c$ of each class. The sparse coefficient vectors of each class thus obtained are then combined from end to end to form one vector of a larger dimension. For example, for speech–music classification system, each speech signal vector $\mathbf{y}_{sp,i}$ from $\mathbf{Y}_{sp,i}$ in the training set is projected on the learned speech $\mathbf{A}_{sp}$ and music $\mathbf{A}_{mus}$ dictionaries separately and the resulting two sparse coefficient vectors each of dimension R^l are combined together to create a sparse coefficient vector of dimension R^{2l}. The same process is also repeated with the music signal

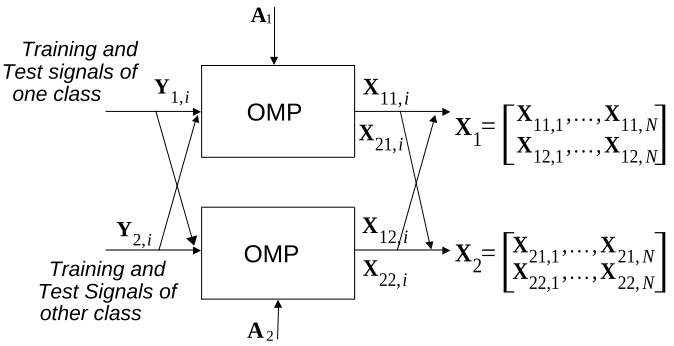


Fig. 3. Extraction of sparse coefficients using OMP. $\mathbf{Y}_{1,i}$ of class-1 signal is mapped to both class specific dictionaries $\mathbf{A}_1$ and $\mathbf{A}_2$ to get their respective sparse coefficients matrices $\mathbf{X}_{11,i}$ and $\mathbf{X}_{12,i}$ which are then concatenated together to form $\mathbf{X}_1$. Same process is repeated for $\mathbf{Y}_{2,i}$.

vectors $\mathbf{y}_{mus,i}$. This process is depicted in Fig. 3. The sparse coefficients of these training signals are used to train the SVM model for signal classification.

The same procedure is used to extract the sparse coefficients of test signals $\mathbf{X}'_{cd,i}$.

2.3. Pooling/sampling of coefficients matrix

Inspired by visual feature extraction methods [35–37], we apply pooling methods to our training and test coefficients matrices to deal with the matrices of different number of columns. Typically, the pooling operation is a sum, an average, a max or any other commutative combination rule. For a sparse coefficient matrix $\mathbf{X}$ extracted from a learned dictionary $\mathbf{A}$, the following pooled feature vectors are obtained by a predefined pooling function

$$\mathbf{z} = \mathcal{F}(\mathbf{X}) \tag{5}$$

where the $\mathcal{F}$ is either a max or an average pooling defined on each row of sparse coefficient matrix $\mathbf{X}$. In case of max pooling, $\mathcal{F}$ is defined as

$$z_p = \max\{|x_{p1}|, |x_{p2}|, \ldots, |x_{pQ}|\} \tag{6}$$

where z_p is the p-th element of $\mathbf{z}$, x_{pq} is the matrix element at p-th row and q-th column of matrix $\mathbf{X}$. For the average pooling, $\mathcal{F}$ is defined as

$$\mathbf{z} = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{x}_q \tag{7}$$

where Q is the total number of coefficients vectors in matrix $\mathbf{X}$. In the max pooling, for each row vector in a matrix $\mathbf{X}$, the element with the maximum value is picked and selected as a representative of that row vector. For the average pooling, the average of all the elements in a row vector of a matrix $\mathbf{X}$ is taken and selected as a representative of that row vector. In this way, each matrix is represented as a single column vector thus reducing the size of data and computational complexity. Hence coefficient matrix $\mathbf{X}_{cd,i}$ is pooled down to vector $\mathbf{z}_{cd,i}$ as shown in Fig. 4. The same pooling operation is applied to the testing signal coefficient matrix $\mathbf{X}'_{cd,i}$ to get $\mathbf{z}'_{cd,i}$.

Pooling is applied to summarize the feature distribution of data of interest into a statistical representation. Hence, here different pooling techniques construct different signal statistics. For sparse codes, the max pooling picks those coefficients values that show maximum contribution from the dictionary atoms while average pooling represents the mean of the dictionary atoms contribution. Further in Section 3.7, we discuss why the max pooling gives better performance as compared to the average pooling. Sparse coding

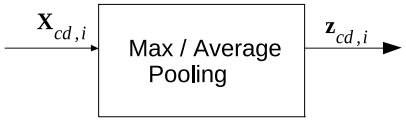


Fig. 4. Max/average pooling of training sparse coefficient matrix $\mathbf{X}_{cd,i}$. This results in the max/average pooled vector $\mathbf{z}_{cd,i}$ of the input matrix.

Fig. 5. Sampling of training sparse coefficient matrix $\mathbf{X}_{cd,i}$ into a size-reduced sampled matrix $\mathbf{X}^r_{cd,i}$.

combined with pooling also reduces the effect of noise in the signals as demonstrated by the experiments.

Pooling is also helpful in making the feature representation compact. In our audio classification system, we use SVM classifier in the classification stage which needs to be trained before the classification of the test data. An alternative way of pooling is to concatenate the column vectors of the coefficient matrices into a single column vector of a larger dimension. In practice, however, the coefficient matrices generated from different training signals have different number of column vectors. As a result, the pooled vectors do not have a uniform length, which makes SVM training less practical. Hence we perform pooling along each row of a coefficient training matrix as indicated in Fig. 4. This transforms the coefficient matrix to a single vector of equal dimension and thus helps to keep the vectors of different sizes of signals compact and in a unified dimension.

Another method for training the classifier is to use all the vectors in the training coefficient matrices as training examples. However, depending upon the number of training signals, the total number of training vectors can become very large which are difficult to be managed by memory-limited computing machines. To cope with this problem, we perform sampling on each signal coefficients matrix to considerably reduce the size of each matrix. Sampling of data vectors is a process of selecting smaller numbers of vectors from training matrix randomly and labeling them with their corresponding signal label. In this way, the reduced set of coefficient vectors represents a training coefficient matrix. Hence the coefficient training matrix $\mathbf{X}_{cd,i}$ is sampled down to the size-reduced matrix $\mathbf{X}^r_{cd,i}$ as shown in Fig. 5, where superscript r shows the reduced size of a matrix. The same process is repeated with the test coefficient matrix $\mathbf{X}'_{cd,i}$ to get $\mathbf{X}'^r_{cd,i}$.

2.4. Signal classification by SVM

Our motive for finding sparse coefficients is to use them for audio classes discrimination in the signal classification stage where the SVM [18] is used as a classifier.

The diagram of the proposed audio signal classification system is shown in Fig. 6. We use the non-linear SVM for our binary classification problem where one class has class label $z_i = +1$ and the other class has class label $z_i = -1$. We use Euclidean distance and radial basis function (RBF) kernel in the SVM. For comparison, we also use linear SVM for one of the experiments of gender classification. The training data points used to define the feature space are vectors $\mathbf{z}_c$ from max/average pooled sparse coefficients matrices $\mathbf{Z}_c$, or vectors $\mathbf{x}^r_c$ from sampled sparse coefficient matrices $\mathbf{X}^r_c$. $\mathbf{z}'_{c,i}$ represents max/average sparse coefficient vector of a test signal. Depending upon the pooling technique, each test signal is classified using its corresponding trained SVM.

For sampled test coefficients, a matrix which represents one signal has a smaller number of vectors than the original signal.

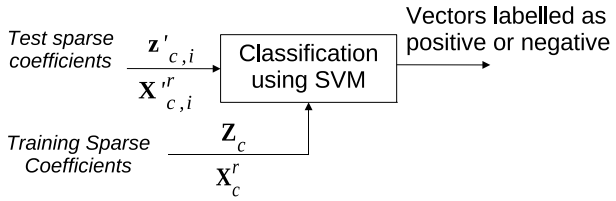


Fig. 6. Sparse coefficients based audio signal classification using SVM. $\mathbf{z}'_{c,i}$ representing a pooled test signal and $\mathbf{X}'^{r}_{c,i}$ representing the sampled test signal are fed to classifier. $\mathbf{Z}_c$ is a matrix containing max/average pooled training vectors and $\mathbf{X}^r_c$ contains sampled training vectors.

Hence the classification decision is made by majority voting of the vectors' labels in a coefficient matrix. If the class labels for the majority of sparse coefficient vectors in a test matrix are positive, it is considered belonging to one class otherwise to the other class.

2.5. Extension to multi-class audio classification

We extend our binary classification system to multi-class audio classification tasks such as speaker identification. With the same classifier setting in one-vs-all fashion, the performance of pooled sparse features is evaluated against the sampled MFCCs to identify C different speakers including male and female. The overall classification accuracy in percentage is calculated as

$$\frac{\sum_{k=1}^{C} \frac{N_{k,a}}{N_{k,t}}}{C} \times 100 \tag{8}$$

where $N_{k,a}$ is the number of correctly classified test signals in one class, $N_{k,t}$ is the total number of test signals in the same class and C is the total number of classes.

3. Experiments

We apply our proposed audio classification system on two binary classification problems and a multi-class problem: speech–music classification, female–male gender classification and speaker identification problem. The datasets, experimental setup and results are presented in the following subsections.

3.1. Datasets

For speech–music classification, 446 different speech signals from TIMIT [38] database are used as training signals which include male and female speakers speaking different sentences with different style and accent. Each signal has a different duration ranging from 1.5 seconds to 5 seconds. Overall, the total duration for 446 speech signals is 22.8 minutes sampled at 16 kHz. Other training data belongs to the music class which is composed of 98 music signals with different notes, taken from the University of Iowa Musical Instruments Database [39]. These music signals are sampled at 44.1 kHz having a total duration of 3.37 minutes. To evaluate the classification performance, additional 125 speech and 49 music signals that were not used during the training process are used in the test stage. The performance comparison between the sparse coefficients and MFCC is shown.

For gender classification, we used the same TIMIT [38] database from which 201 female speech signals with a total duration of 10.3 minutes and 245 male speech signals with a total duration of 12.5 minutes are chosen for training, and 40 female speech and 50 male speech signals for testing. The training and testing data do not overlap with each other.

A subset of TIMIT corpus is selected for speaker identification of 5 speakers and 10 utterances (sentences) per speaker, resulting in a total of 50 utterances. For different number of utterances per speaker, we perform classification in such a way that training and testing examples do not overlap with each other. Sparse coefficients for training and testing data are extracted as described in Section 2 and its classification performance is compared with that of the sampled MFCC and DicClassifer [25].

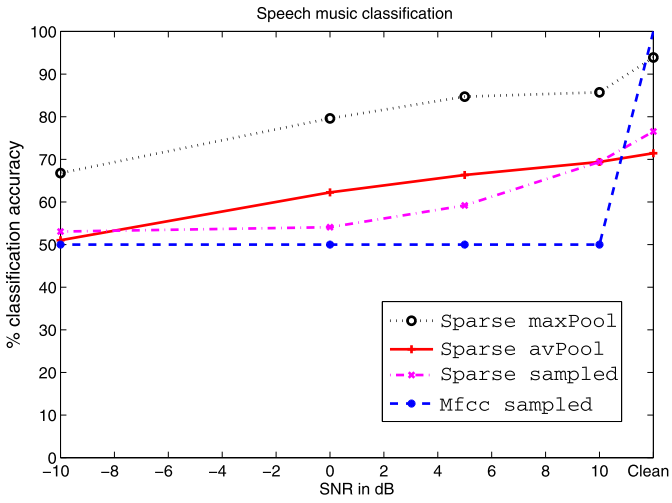


Fig. 7. Speech–music classification for noisy testing data with SNR changing from −10 dB to 10 dB based on clean training data. Results for clean testing data are also shown.

3.2. Setup

In two binary classification problems, experiments are performed using clean as well as noisy training and testing data. Different levels of white Gaussian noise with zero mean and unit variance is added to the training and testing data with the signal to noise ratio (SNR) ranging from 10 dB to −10 dB. We further explore the effect of various dictionary size on the classification performance.

By using K-SVD, class specific dictionaries of size 256×1000 and 256×700 are learned for speech–music and female–male classification tasks respectively. For speaker identification, speaker specific dictionaries for various training utterances per speaker are learned, each of size 320×320.

The sparse coefficient vectors of the training and test signals are calculated on a frame-by-frame basis. Each frame is mapped to speech and music dictionaries separately which results in two sparse coefficient vectors, each of dimension 1000 having maximum 13 non-zero values. These two sparse coefficient vectors per frame are combined together to obtain a single coefficient vector of dimension 2000 with maximum 26 non-zero values. The resulting coefficient matrices per signal per class are max pooled, average pooled and sampled to get training and testing vectors for the SVM classifier.

In matrix sampling operation, one out of 10 column vectors is picked up randomly and selected as a representative of the 10 column vectors. These frames are also used to get the MFCC coefficients of the training signals with each having a dimension of 13 followed by normalization of values between 0 and 1. Since pooling operations are only applicable to codebook/dictionary based features, only sampling operation is applied to MFCC matrices. For sampled feature vectors, classification decision is made based on majority voting.

3.3. Results

3.3.1. Classification using clean training data

Using the clean training data, the overall classification accuracy with varying SNRs of testing data is given in Fig. 7 using max-

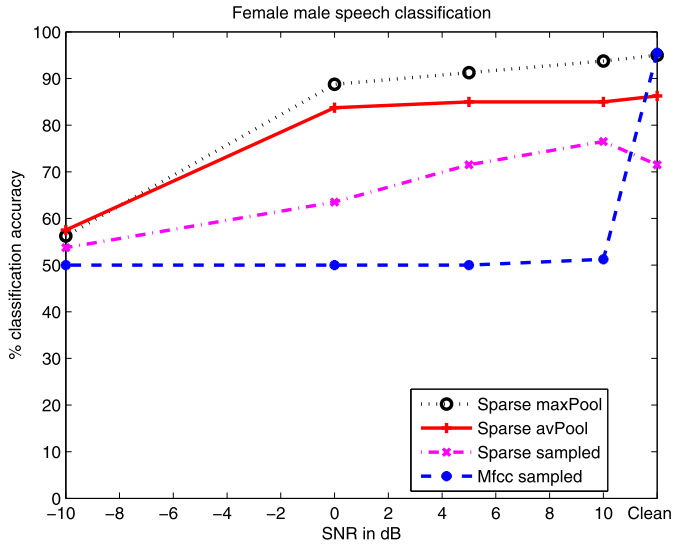


Fig. 8. Female–male speech classification for noisy testing data with SNR changing from −10 dB to 10 dB based on clean training data. Results for clean testing data are also shown.

pooled, average-pooled and sampled sparse coefficients as well as sampled MFCCs along with the results for clean testing signals.

Fig. 7 clearly shows supremacy of sparse coefficients over MFCCs as good features for classification. For added noise, max pooled sparse coefficients perform better as compared to other pooled or sampled sparse coefficients as well as sampled MFCCs. This shows that for each signal, the highest value of the sparse coefficients exhibits the discerning signal feature for classification. Both the average pooled and sampled sparse coefficients perform better as compared to the sampled MFCC coefficients for the noisy signals. The only exception when MFCC outperforms sparse coefficients is for the clean test signals (without any noise). However the trade-off lies in its increased computational complexity as sampled MFCC matrices are used as training and testing examples as compared to the pooled sparse coefficients matrices. Moreover, most audio signals of practical interest have some noise in them which implies that sparse coefficients are better options for speech–music classification in practice.

Fig. 8 shows performance for gender speech classification. Again in this case, the max-pooled sparse coefficients give the best performance followed by average pooled and then sampled sparse coefficients. This shows that even with a higher computational complexity, the performance using MFCC is poorer as compared to that using sparse coefficients.

3.3.2. Classification using noisy training data

To evaluate the classification robustness based on different features, we also perform classification using noisy training data with SNR varying from 10 dB to −10 dB. Figs. 9(a)–(d) show the overall speech–music classification performance for noisy training data with SNR of 0 dB, 5 dB, 10 dB and −10 dB. In all these figures, the max-pooled sparse coefficients show more robust noise rejection capability as compared to other coefficients. Sampled MFCCs give the best classification performance when the SNR of training signals is similar to that of test signals. Beyond that specific SNR range, MFCC performance degrades. The classification accuracy variance based on max-pooled and average-pooled sparse coefficients is lower as compared to sampled MFCCs and sparse coefficients. This shows that the pooled sparse coefficients are robust features against noise in general and max-pooled sparse coefficients in particular.

Figs. 10(a)–(d) show the accuracy of female–male speech classification with noisy training data of 0 dB, 5 dB, 10 dB and −10 dB respectively. These figures show that the pooled and sampled sparse coefficients are better for female–male speech classification as compared to the sampled MFCCs. Particularly the low variance of max pooled sparse coefficients based classification results shows their good robustness to noise.

3.3.3. Effect of dictionary size on sparse coefficients based classification

The results we have shown so far are based on the dictionary size of 1000 for speech–music classification and 700 in male–female speech classification. Fig. 11 shows audio classification based on different dictionary sizes for noisy as well as clean testing data with clean training data. The SNR for noisy testing data changes from −10 dB to 10 dB. Figs. 11(a) and (b) show speech–music classification results based on max-pool and average-pool sparse coefficients, respectively with variable dictionary size changing from 256 to 1300 while Figs. 11(c) and (d) show female–male speech classification results. Fig. 11 shows that changing the dictionary size does not change noticeably the classification performance for speech–music as well as gender speech classification. For female–male speech classification, max-pooled sparse coefficients give better performance for SNR = −10 dB with reduced dictionary size while average pooled sparse coefficients show comparable classification accuracy for a larger dictionary size. The sparsity level in the case of both dictionary sizes is fixed i.e. 13. However, an overall trend is that the dictionary size does not affect the classification performance much.

3.4. Multi-class classification

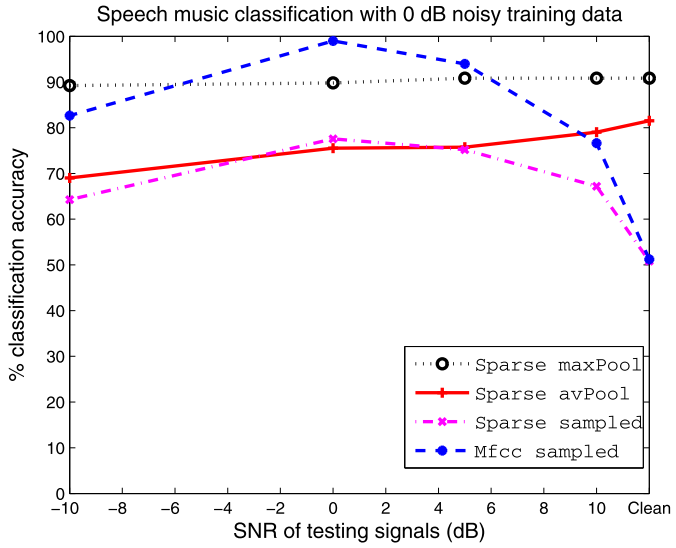
We have also investigated the performance of using pooled sparse coefficients for a multi-class problem of speaker identification. The classification results are shown in Table 1. For different number of utterances per speaker, the max pooled sparse features outperform the MFCC features. This shows that in multi-class case too, the max pooled sparse features are better than the MFCCs.

3.5. Classification with linear SVM

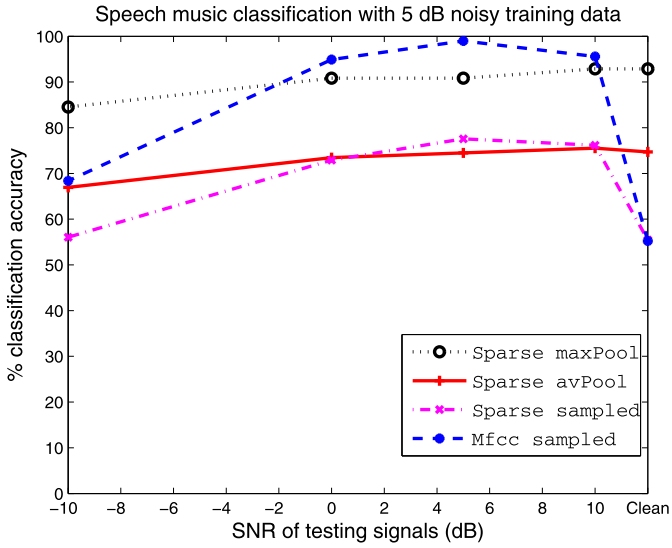
The classification results shown so far were computed using non-linear SVM with RBF kernel. We also show some results obtained using linear SVM. To this end, we repeat the female–male speech classification experiments performed in Fig. 8 and the multi-class classification experiments performed in Table 1, by replacing the non-linear SVM with the linear SVM. The results for gender classification are shown in Fig. 12. It appears that the classification accuracy using the linear SVM is similar to that of the non-linear SVM. By comparing Tables 1 and 2, the same trend for speaker identification can also be seen.

3.6. Comparison with DicClassifier

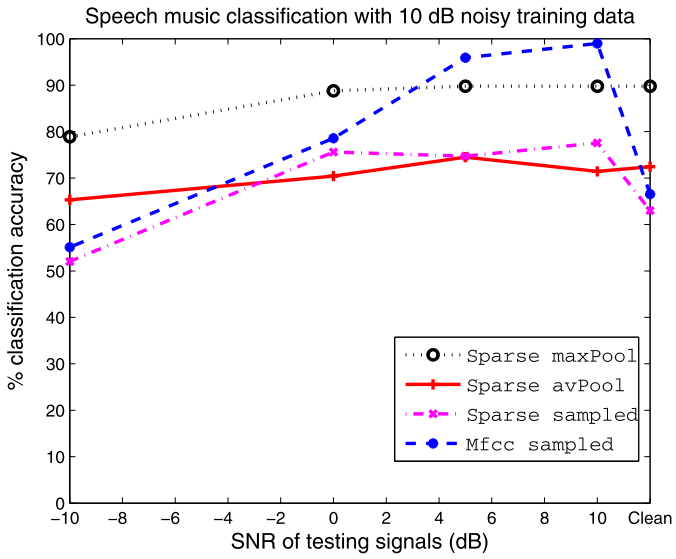
We also compared sparse max pool features with the work presented in [25]. For comparison purposes, we named the algorithm as DicClassifier. This method takes raw audio data, converts them to matrix form, extract linear predictive coding (LPC) features from them, learns LPC-dictionaries for each class and then uses those class specific LPC-dictionaries as classifier for the classification of test signals. The test signals are also converted to LPC coefficients and then projected on the class specific LPC-dictionaries for classification. To make a fair comparison of our work with that of DicClassifier, we learned 13 LPC features from audio data frames and then learned dictionaries of the same size as that of max pooled sparse coefficients dictionaries. 13 LPC coefficients were chosen because we used the same number of coefficients for MFCC and sparse coefficients.



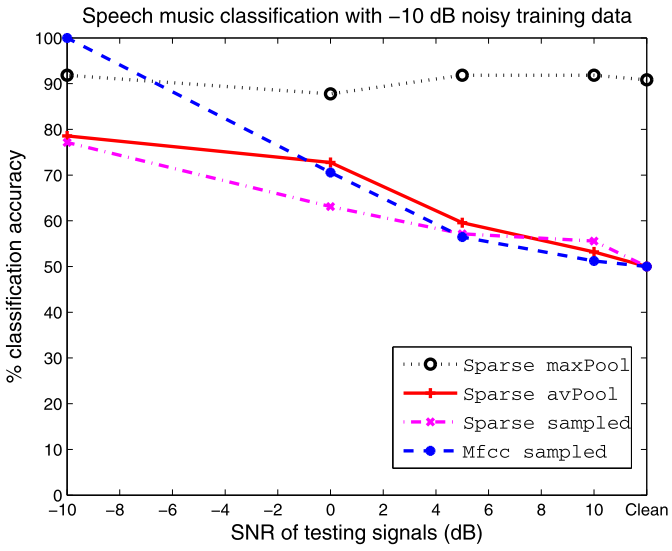
(a) 0 dB noisy training signals.



(b) 5 dB noisy training data.



(c) 10 dB noisy training data.



(d) -10 dB noisy training data.

Fig. 9. Speech–music classification with training and testing data both distorted by additive white Gaussian noise with SNR changing from −10 dB to 10 dB for the testing data. Results for clean testing data are also shown. (a) Training data SNR = 0 dB. (b) Training data SNR = 5 dB. (c) Training data SNR = 10 dB. (d) Training data SNR = −10 dB.

Table 1
Classification performances for identification of 5 speakers for different features using non-linear SVM with different number of utterances per speaker.

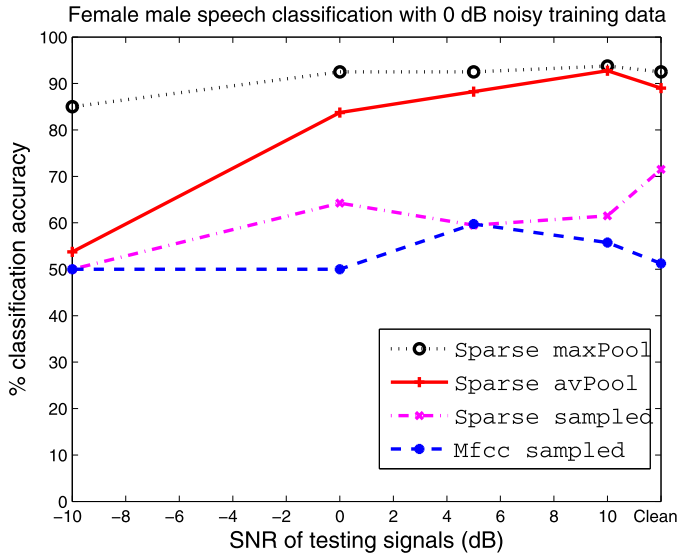
# training utterances per speaker	MFCC	Sparse maxPool	Sparse avPool
2	47.96%	**74.29%**	45.71%
5	53.42%	**72%**	68%
8	50.88%	**80%**	50%

Table 2
Classification performances for the identification of 5 speakers for different features using the linear SVM with different number of utterances per speaker.

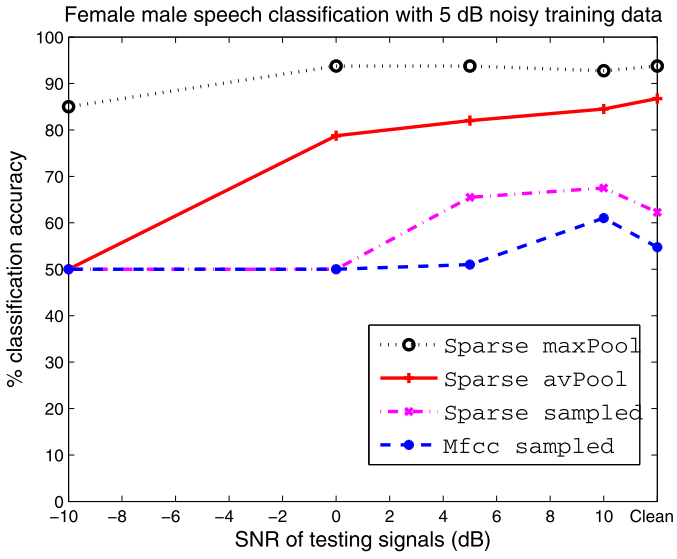
# training utterances per speaker	MFCC	Sparse maxPool	Sparse avPool
2	46.28%	**71.43%**	45.71%
5	49.81%	**72%**	32%
8	48.51%	**80%**	50%

Instead of learning features from raw audio data, DicClassifier learns dictionaries of higher level features of audio signals. This is similar to the work as described in [2] where the combination of MFCC and sparse features were used for the classification of environmental sounds which naturally improves the classification performance. However our work is to tweak with learned sparse features of raw audio data by using different pooling techniques to investigate their performance in comparison with those of non-pooled sparse coefficients and conventional audio features like MFCC.

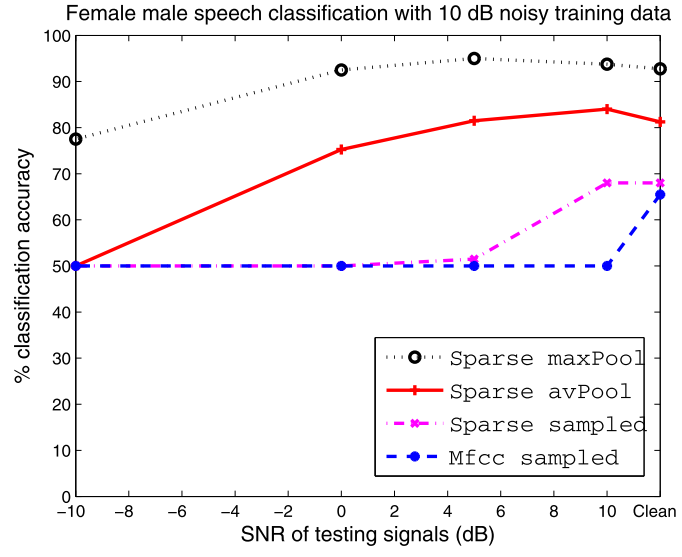
The classification performance of DicClassifier using LPC as well as raw data is shown in Table 3. For a small number of training utterances, sparse max pooled coefficients outperform DicClassifier. However when using 5 and 8 training utterances for each speaker, DicClassifier performs better only when LPC features are used as input. If the raw data is used, DicClassifier has far poorer performance as compared to the use of pooled sparse features. This shows that if the max pooled sparse features are combined with high level features of audio data like LPC or MFCC, the classification performance can be considerably improved.



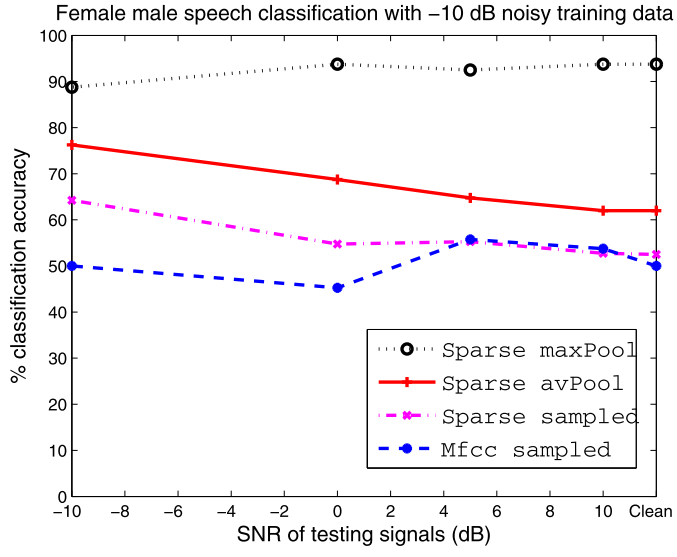
(a) 0 dB noisy training data.



(b) 5 dB noisy training data.



(c) 10 dB noisy training data.



(d) -10 dB noisy training data.

Fig. 10. Female–male speech classification with training and testing data both distorted by additive white Gaussian noise with SNR changing from −10 dB to 10 dB for testing data. Results for clean testing data are also shown. (a) Training data SNR = 0 dB. (b) Training data SNR = 5 dB. (c) Training data SNR = 10 dB. (d) Training data SNR = −10 dB.

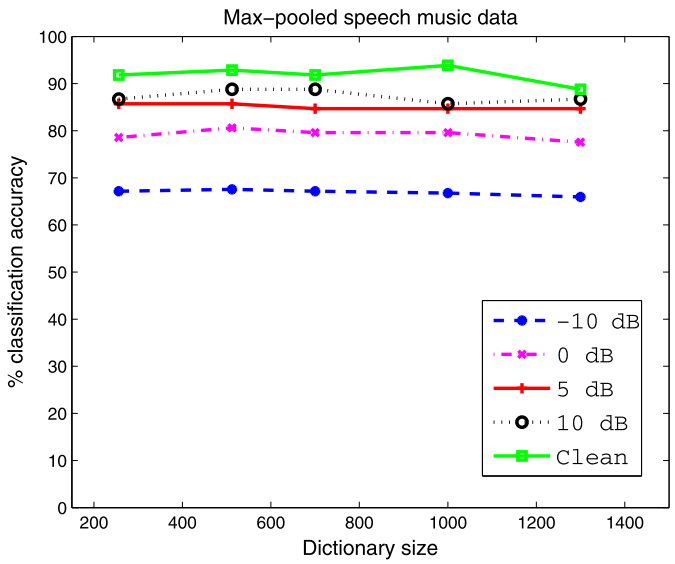
3.7. Further discussion

Results suggest that learned sparse coefficients show promising characteristics as feature representative of audio signals. Particularly, max pooled sparse coefficients give excellent performance for a large range of noisy training and testing data. This is because an overcomplete dictionary gives rise to sparse coefficients, whose maximum value represents the response of a dictionary atom showing maximum contribution in defining the signal feature. In case of feature sampling, many dictionary atoms in speech as well as music may be similar. Hence, while sampling the training coefficient matrices to obtain the subset of the original coefficient vectors, most of the coefficient values may represent similar dictionary atoms, though from different classes. This confuses the classifier during the training process and thus degrades the overall classification accuracy. Moreover, this sampling process may or may not represent the dictionary elements with the strongest contribution towards signal's feature representation. It seems that with sampling, our classification accuracy should improve as we are selecting more vectors to represent signal features. However, it shows degradation in classification accuracy even with an increase in computational complexity and memory requirement.

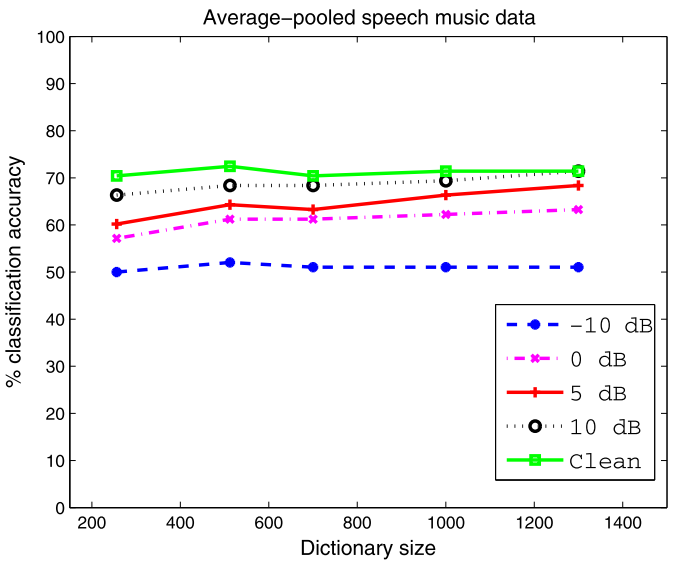
Table 3
Classification performances for the identification of 5 speakers using DicClassifier.

# training utterances per speaker	DicClassifier	
	Raw audio	LPC
2	28.57%	65.71%
5	36%	84%
8	30%	90%

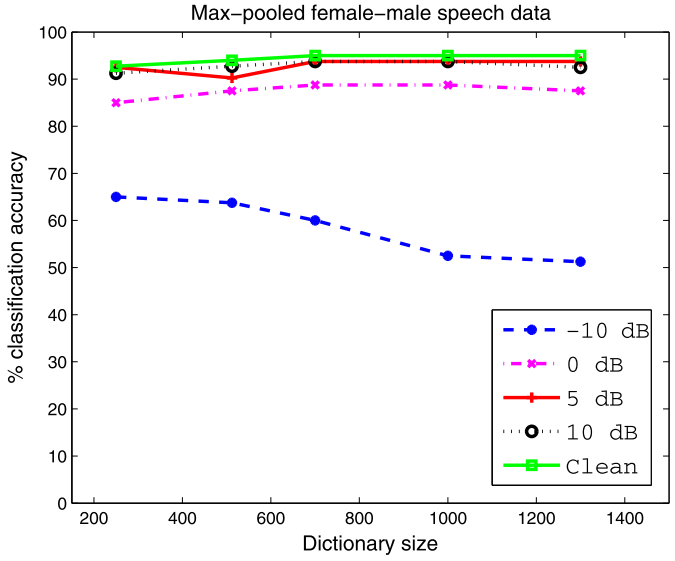
For speech–music classification with noisy training data, MFCCs show better performance when the noise level of training and testing signals is equal while such behavior is not shown in female–male speech classification. This shows that MFCCs are good in discriminating those classes which are highly separable as in the case of speech–music class. However, when classes are overlapping like female–male speech class, their performance degrades.



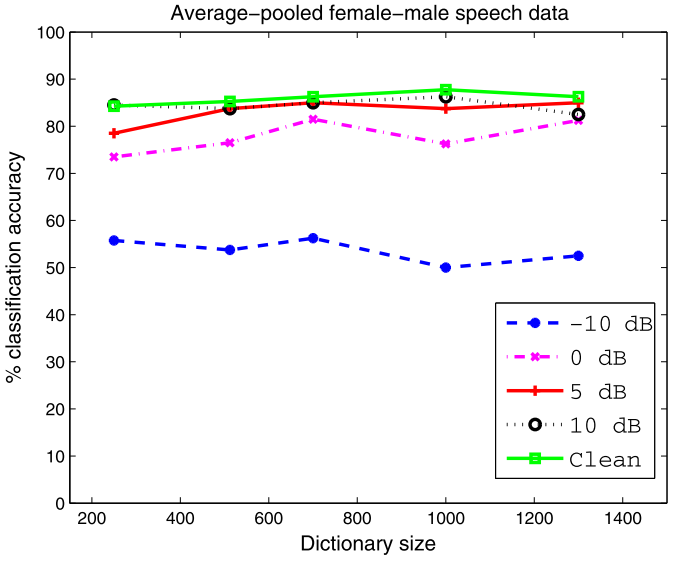
(a) Max-pooled sparse coefficients classification for speech music data.



(b) Average-pooled sparse coefficients classification for speech music data.



(c) Max-pooled sparse coefficients classification for female-male speech.



(d) Average-pooled sparse coefficients classification for female-male speech.

Fig. 11. Audio classification based on variable dictionary size for noisy as well as clean testing data using clean training data. The SNR for noisy testing data changes from −10 dB to 10 dB.

The max-pooled sparse coefficients show better performance even when the classes are overlapping. Average pooling for sparse coefficients gives relatively lower performance as compared to max pooled coefficients. This means that average pooling reduces the contribution of the most active dictionary elements towards signal features. This dilution of most active dictionary elements gives poorer performance.

The overall classification results based on sparse coefficients seem to be less affected by the dictionary size. This shows that pooled and sampled sparse coefficients always select best dictionary atoms for sparse representation irrespective of dictionary size.

Another benefit of pooling operation is its robustness to noise. Superior performance of pooling not only shows the dilution of noise added to the coefficient elements but also the rejection of noisy elements, particularly in the case of max pooling.

In addition to the advantages of pooling operation discussed above, the reduction of the size of each training and testing signal's coefficient matrix to a vector drastically decreases the overall computational complexity of the whole classification process.

4. Conclusion

We have presented a method of using learned dictionaries to extract signal features for speech–music and female–male speech classification, as well as speaker identification. We learned different dictionaries with each representing one class of signal. Using those dictionaries, we calculated the sparse coefficients of each class. These sparse coefficients were further nurtured by using pooling and sampling techniques. We found that those sparse coefficients were very good representatives of signal features that can be used for speech discrimination and speaker identification. Par-

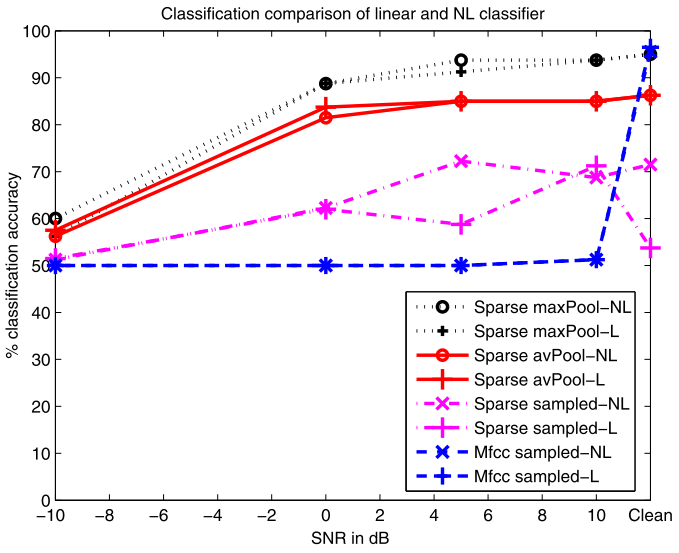


Fig. 12. Comparison of classification performance for gender classification with linear and non-linear SVM. L and NL in legend represent the linear and non-linear SVM respectively.

ticularly, the max pooled sparse coefficients vectors best described the signal features. Our results show that the pooled sparse coefficients outperform the MFCC features for the task of audio classification, particularly for noisy data and overlapping classes. Moreover, as the pooling technique summarizes a coefficient matrix in to a vector, the computational complexity of the classification process is drastically reduced which makes it potentially useful to be considered for future online applications.

Acknowledgments

The authors are grateful to the reviewers for their valuable comments for improving their paper. This research is funded in part by International Islamic University, Islamabad, Pakistan under the research development program, and the Engineering and Physical Sciences Research Council of the UK (grant numbers EP/H050000/1 and EP/H012842/1).

References

[1] J. Saunders, Real-time discrimination of broadcast speech/music, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, 1996, pp. 993–996.
[2] S. Chu, S. Narayanan, C.-C. Jay Kuo, Environmental sound recognition using MP-based features, in: International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 1–4.
[3] E. Wold, T. Blum, D. Keislar, J. Wheaton, Content-based classification, search and retrieval of audio, IEEE Trans. Multimed. 28 (1996) 27–36.
[4] C. West, S. Cox, Features and classifier for the automatic classification of musical audio signals, in: International Conference for Music Information Retrieval, 2004, pp. 531–537.
[5] M.J. Carey, E.S. Parris, H. Lloyd-Thomas, A comparison of features for speech, music discrimination, in: International Conference on Acoustics, Speech and Signal Processing, vol. 1, 1999, pp. 149–152.
[6] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Prentice Hall, 1978.
[7] T. Zhang, C.J. Kuo, Hierarchical classification of audio data for archiving and retrieving, in: International Conference on Acoustics, Speech and Signal Processing, vol. 6, 1999, pp. 3001–3004.
[8] J.T. Foote, Content-based retrieval of music and audio, in: Proceedings of SPIE, 1997, pp. 138–147.
[9] K. El-Maleh, M. Klein, G. Petrucci, P. Kabal, Speech/music discrimination for multimedia applications, in: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 6, 2000, pp. 2445–2446.
[10] E. Scheirer, M. Slaney, Construction and evaluation of a robust multifeature speech/music discriminator, in: International Conference on Acoustics, Speech and Signal Processing, vol. 2, 1997, pp. 1331–1334.
[11] M. Mckinney, J. Breebaart, Features for audio and music classification, in: International Symposium on Music Information Retrieval, 2003, pp. 151–158.
[12] E. Zwicker, H. Fastl, Psychoacoustics: Facts and Models, Springer Series on Information Science, 1999.
[13] G. Williams, D.P.W. Ellis, Speech/music discrimination based on posterior probability features, in: Proceedings of the 6th European Conference on Speech Communication and Technology, 1999, pp. 687–690.
[14] L. Lu, H.J. Zhang, H. Jiang, Content analysis for audio classification and segmentation, IEEE Trans. Speech Audio Process. 10 (2002) 504–516.
[15] Z. Liu, J. Huang, Y. Wang, I.T. Chen, Audio feature extraction and analysis for scene classification, in: IEEE First Workshop on Multimedia Signal Processing, 1997, pp. 343–348.
[16] J.G.A. Barbedo, A. Lopes, A robust and computationally efficient speech/music discriminator, J. Audio Eng. Soc. 54 (2006) 571–588.
[17] J. Ajmera, I. McCowan, H. Bourlard, Speech/music segmentation using entropy and dynamism features in a HMM classification framework, Speech Commun. 40 (2003) 351–363.
[18] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods), Cambridge University Press, 2000.
[19] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (2006) 4311–4322.
[20] M. Yaghoobi, T. Blumensmith, M.E. Davies, Dictionary learning for sparse approximations with the majorization method, IEEE Trans. Signal Process. 57 (2009) 2178–2190.
[21] Y. Li, S. Amari, A. Cichocki, D.W.C. Ho, X. Shengli, Underdetermined blind source separation based on sparse representation, IEEE Trans. Signal Process. 54 (2006) 423–437.
[22] S. Scholler, H. Purwins, Sparse approximations for drum sound classification, IEEE J. Sel. Top. Signal Process. 5 (2011) 933–940.
[23] R. Grosse, R. Raina, H. Kwong, A.Y. Ng, Shift-invariant sparse coding for audio classification, in: Conference on Uncertainty in Artificial Intelligence, 2007, pp. 149–158.
[24] H. Lee, Y. Largman, P. Pham, A.Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, Adv. Neural Inf. Process. Syst. 22 (2009) 1096–1104.
[25] C. Rusu, Classification of music genres using sparse representations in overcomplete dictionaries, J. Control Eng. Appl. Inf. 13 (2011) 35–42.
[26] K. Delgado, J.F. Murray, B.D. Rao, K. Engan, T. Lee, T.J. Sejnowski, Dictionary learning algorithms for sparse representations, Neural Comput. 15 (2003) 349–396.
[27] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: A strategy employed by v1, Vis. Res. 37 (1997) 3311–3325.
[28] K. Engan, S. Aase, J. Hakon-Husoy, Method of optimal directions for frame design, in: International Conference on Acoustics, Speech and Signal Processing, vol. 5, 1999, pp. 2443–2446.
[29] J. Tropp, Greed is good: Algorithmic results for sparse approximation, IEEE Trans. Inform. Theory 50 (2004) 2231–2242.
[30] M. Mallat, Z. Zhang, Matching pursuit in a time–frequency dictionary, IEEE Trans. Signal Process. 41 (1993) 3397–3415.
[31] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[32] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B 58 (1994) 267–288.
[33] R.F. Gorodnitsky, B.D. Rao, A recursive weighted minimum norm algorithm: Analysis and applications, in: International Conference on Acoustics, Speech and Signal Processing, vol. III, 1993, pp. 456–459.
[34] T. Blumensath, M.E. Davies, Gradient pursuits, IEEE Trans. Signal Process. 56 (2008) 2370–2382.
[35] Y.-L. Boureau, J. Ponce, Y. Lecun, A theoretical analysis of feature pooling in visual recognition, in: International Conference on Machine Learning, 2010, pp. 111–118.
[36] P. Koniusz, K. Mikolajczyk, Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match, in: International Conference on Image Processing, 2011, pp. 661–664.
[37] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.
[38] J.S. Garofolo, et al., TIMIT Acoustic–Phonetic Continuous Speech Corpus, Linguistic Data Consortium, Philadelphia, 1995.
[39] The University of Iowa Musical Instruments Samples Database, online: http://theremin.music.uiowa.edu, 2011.

Syed Zubair received his B.Sc. and M.Sc. in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan. Currently, he is pursuing his Ph.D. in the Centre for Vision, Speech and Signal Processing, University of Surrey, UK.

Dr. Fei Yan is a research fellow in the Centre for Vision, Speech and Signal Processing (CVSSP) at the University of Surrey. His research interests include machine learning and computer vision, in particular, kernel methods, object recognition, and object tracking. He has publications in major machine learning and computer vision conferences and journals, including ICDM, ECML, CVPR, PAMI, JMLR.

Wenwu Wang is a Lecturer at the Centre for Vision, Speech and Signal Processing, University of Surrey, since May 2007. Prior to this, he was a Postdoctoral Research Associate at King's College London (from May 2002 to December 2003) and Cardiff University (from January 2004 to April 2005).

He also worked in UK industry, first as a DSP Engineer at Tao Group Ltd (now Antix Labs Ltd) (from May 2005 to August 2006), then as an R&D engineer at Creative Labs (from September 2006 to April 2007). During spring 2008, he has been a visiting scholar at the Perception and Neurodynamics Lab and the Center for Cognitive Science, The Ohio State University. He is currently a member of the MOD University Defense Research Centre in Signal Processing and the BBC Audio Research Partnership. He obtained the PhD degree in April 2002 from Harbin Engineering University, China.

His current research interests include blind signal processing, sparse signal processing, audio-visual signal processing, machine learning and perception, and machine audition (listening). He is a Senior Member of the IEEE, and belongs to the IEEE Signal Processing, Computational Intelligence, and Circuits and Systems Societies.

He was an Area Chair of the 2012 European Signal Processing Conference, a Track Chair and Publicity Co-Chair of 2009 IEEE Statistical Signal Processing Workshop, Program Co-Chair of the 2009 IEEE Global Congress on Intelligent Systems. He has been a Session Chair for numerous conferences including ICASSP 2012 and EUSIPCO 2012.

He won the DSTL Best Solution Award (with Qingju Liu) in the DSTL Challenge Workshop for the signal processing challenge "under-sampled signal recognition" in 2012, the Best Student Paper Award nomination (with Qingju Liu) at the 9th International Conference on Latent Variable Analysis and Signal Separation in 2010, and the Hot Paper (feature article) of the Wiley/IEEE worldwide advert for publications in signal and image processing in 2008.