Source Microphone Recognition Aided by a Kernel-Based Projection Method

Yuechi Jiang, *Student Member, IEEE*, and Frank H. F. Leung, *Senior Member, IEEE*

***Abstract*— Microphone recognition aims at recognizing different microphones based on the recorded speeches. In the literature, Gaussian Supervector (GSV) has been used as the feature vector representing a speech recording, which is obtained by adapting a universal background model (UBM). However, it is not clear how the performance of the GSV will be affected by the number of mixture components in the UBM. Besides, the raw GSV obtained from a speech recording contains both the microphone response information and the speech information, meaning that the raw GSV can be quite noisy as the feature vector for microphone recognition. In this paper, we investigate how GSV will be affected by the UBM and other parameters during the calculation of the GSV. In addition, in order to improve the quality of the raw GSV, we propose a kernel-based projection method to be applied to the raw GSV. This projection method maps the raw GSV onto another dimensional space. It is expected that in the projected feature space, the microphone response information and the speech information can be separated into different dimensions, meaning that the projected GSV should be better as the feature vector for microphone recognition compared to the raw GSV. Two classifiers that have been used in the literature, namely linear support vector machine (SVM) and sparse representation-based classifier (SRC), are employed to compare the performance of the raw GSV and the projected GSV. The experimental results demonstrate that the projected GSV can outperform the raw GSV no matter using linear SVM or SRC as the classifier, which shows the effectiveness of the projection method.**

***Index Terms*— Kernel-based projection, linear support vector machine, microphone recognition, sparse representation based classifier.**

I. Introduction

THE speech content conveyed by a speech recording is usually the most important information. However, some other information carried by a speech recording can also be useful. For example, the speech recording may embed the information about the recording device, such as the microphone, that is used to record the speech. A speech recording may also embed the recording date information; for example, if the speech is recorded near a power grid, it will embed the Electric Network Frequency (ENF) signal, which may be used as a time stamp [1]. This extra information can be very useful, or even used as court evidence if the speech is not tampered. In this paper, the focus is on the microphone information (i.e. recording device information) embedded in the speech signal, and the objective is to recognize the recording microphone based on the recorded speech.

In [2], Naïve Bayes was employed as the classifier to do microphone classification. Although the classification accuracy was not good, the research showed the possibility of recognizing the recording microphone based on the recorded audio. After that, different feature vectors have been proposed to capture the device information, and different classifiers have been employed to identify the recording device. For example, in [3], two classifiers were employed: one is the decision tree, and the other is a linear regression model. After carefully fusing the results from the two classifiers, the classification accuracy can be better than that using a single classifier. In [4] and [5], Gaussian Mixture Model (GMM) was employed as the classifier in identifying 16 different microphones, and the accuracy was good on three datasets.

Since speech recordings are usually of different lengths, it is necessary to first divide a speech recording into a sequence of frames to obtain a sequence of frame-level features, and then form a single feature vector from the frame-level features. A good choice of the frame-level feature is the Mel-frequency Cepstral Coefficient (MFCC) vector, which has been widely used in speech recognition and speaker verification. In microphone recognition, some other frame-level features have also been evaluated, such as the Multi-taper MFCC [5], or the Linear Prediction Cepstral Coefficient (LPCC) [4]. Directly using the spectrum instead of using MFCC or LPCC has also been shown to perform well in telephone handset identification, such as the Random Spectral Features (RSFs) [6], the Sketches of Spectral Features (SSFs) [7], and the Labeled Spectral Features (LSFs) [8].

On how to form a single feature vector from a sequence of frame-level features, one way is simply to average the frame-level features, and another way is to construct a Gaussian Supervector (GSV). GSV has been successfully applied to speaker recognition and verification, and is also shown to give good performance in microphone recognition and telephone handset recognition [9]. In fact, besides microphone recognition, GSV has also been applied to other recording device recognition tasks, such as mobile phone identification [10], mobile phone verification [11] and comparison [12].

Regarding the classifier, Support Vector Machine (SVM) has been shown to be good in microphone recognition [9]

Manuscript received March 5, 2018; revised May 29, 2018 and February 28, 2019; accepted March 26, 2019. Date of publication April 15, 2019; date of current version June 27, 2019. This work was supported by The Hong Kong Polytechnic University under Grant RUG7. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefano Tubaro. *(Corresponding author: Yuechi Jiang.)*

The authors are with the Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: yuechi.jiang@connect.polyu.hk; frank-h-f.leung@polyu.edu.hk).

Digital Object Identifier 10.1109/TIFS.2019.2911175

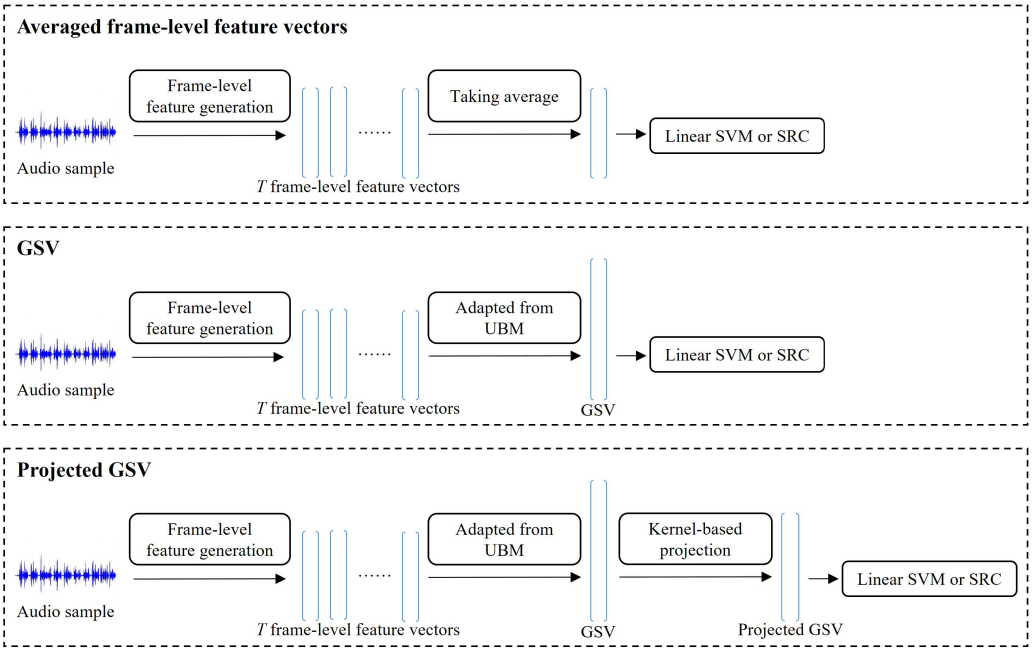


Fig. 1. Overview of different feature vectors (averaged frame-level feature, raw GSV, projected GSV) and different classifiers (linear SVM, SRC).

and telephone handset recognition [6]–[10]. In addition, Sparse Representation based Classifier (SRC) has also been found to be good in telephone handset recognition [6]–[8], with its performance comparable to that of SVM. SRC has also been applied to mobile phone verification [11] and comparison [12].

While the aforementioned studies are focusing on closed-set microphone recognition using clean audio recordings, some studies try to deal with noisy audio recordings [13] or open-set microphone recognition [14]. Instead of identifying different models of microphones, [15] tried to identify two microphones with the same model. Besides using some machine learning techniques in microphone identification, [16] modeled the response of a microphone as a nonlinear system. Microphone classification was also shown to be useful in audio tampering detection [17].

It is known that a recorded speech signal can be regarded as the convolution of the source speech signal and the recording device impulse response [18]. For microphone recognition, the device information is useful whereas the speech information is not only useless but also interfering. However, it is inherently difficult to separate the device information from the speech information, because both the device information and the speech information are unknown beforehand. Thus some studies try to make use of the near-silence segments of the recorded audio signal [19], or non-speech segments [20] or noise signal [21], to extract features. These segments usually contain a small amount of speech information, thus the features extracted from these segments may be less interfered by the speech information. However, these near-silence signals are unstable since they are quite noise-like, and insufficient to train a good classifier if the audio signal is filled with speech information. Instead, in [22], the low-energy segments (i.e. near-silence segments) and the high-energy segments were combined in a weighted manner, and the Weighted Support Vector Machine (WSVM) was employed.

In this paper, to carry out microphone recognition, GSV is used as the feature vector representing a speech recording, and both linear SVM and SRC are used as the classifier. GSV is believed to be a good feature vector, as it is of high dimensionality, and thus can capture enough information about the recording device. However, this device information is usually interfered by other prominent information, such as the speech information. In order to separate the source information (i.e. speech signal) and the device information (i.e. the impulse response of the recording device), a kernel-based projection method is proposed. This method projects the raw GSV onto another dimensional space. It is expected that in the projected space, the source information and the device information can be well separated, meaning that the projected GSV should be better oriented towards microphone recognition than the raw GSV. This projection method is inspired by the Nuisance Attribute Projection (NAP) method [23], [24] applied to speaker recognition and verification. It will be shown later that no matter using linear SVM or SRC, the projected GSV outperforms the raw GSV. An overview of different feature extraction methods and classifiers is depicted in Fig. 1.

This paper is organized as follows. In Section II, the formulation of GSV is described. In Section III, the kernel-based projection method is described. In Section IV, sparse representation and SRC are briefly explained. In Section V, the microphone speech corpus is briefly described. In Section VI, experimental results on how GSV will be affected by the UBM as well as other parameters, and the comparison of the raw GSV and the projected GSV employing linear SVM and SRC

as the classifier, are presented and discussed. In Section VII, the conclusion is drawn.

II. Frame-Level Feature and Gaussian Supervector

A. Frame-Level Feature

The popular MFCC vector is used as the frame-level feature, whose detailed formulation can be found in [25]. Hamming window with 50ms frame length and 10ms frame shift is used to obtain the short-time frames. Then 48 Mel-scale triangular filters are used to filter the frequency spectrum of a frame, followed by the Discrete Cosine Transform (DCT) applied to the filtered spectrum. Finally, the first 24 DCT coefficients (starting from the second coefficient) are used to form a 24-dimension MFCC vector, which excludes the energy coefficient.

B. Gaussian Supervector

GSV is constructed through adapting a Universal Background Model (UBM), which is namely a GMM. The UBM is used to reflect the general statistics of a large number of speech recordings, while a GSV corresponding to one speech recording combines the statistics from this specific speech recording and the statistics from the UBM. Given a set of speech recordings for the UBM construction, each speech recording is first divided into short-time frames. Then, the short-time frames are transformed into MFCC vectors. These MFCC vectors are used to construct the UBM, using the mixture splitting technique [26] and the Expectation-Maximization (EM) algorithm [27], [28]. The mixture splitting and EM retraining process is carried out as follows. Suppose there have already been an m-mixture UBM with the parameter set denoted as $\theta_m = \{\pi_i, \mu_i, \sigma_i | i = 1, 2 \ldots m\}$, where π_i, μ_i and σ_i are the weight, the mean vector and the standard deviation vector (assuming a diagonal covariance matrix in the UBM) for the i-th Gaussian mixture component in the UBM. Having the parameter set θ_m, in order to build a $2m$-mixture UBM, two new parameter sets are first constructed from θ_m, denoted as $\theta_m^{(1)} = \{\frac{\pi_i}{2}, \mu_i + 0.2\sigma_i, \sigma_i | i = 1, 2 \ldots m\}$ and $\theta_m^{(2)} = \{\frac{\pi_i}{2}, \mu_i - 0.2\sigma_i, \sigma_i | i = 1, 2 \ldots m\}$ respectively. Then the initial parameter set for the $2m$-mixture UBM is the combination of these two sets, denoted as $\theta_{2m} = \theta_m^{(1)} \cup \theta_m^{(2)}$. θ_{2m} is then retrained using EM algorithm. To build a UBM with M (which is assumed to be a power of 2) Gaussian mixture components, a UBM with a single Gaussian mixture component is split for $\log_2 M$ times, and each time the number of Gaussian mixture components is doubled.

Suppose there have already been an M-mixture UBM, then for a training or testing speech recording, a sequence of MFCC vectors $\{z_1, z_2 \ldots z_T\}$ are first calculated, where T is the total number of MFCC vectors obtained from this speech recording. Then, GSV is calculated using (1) $\sim$ (4). In (1) $\sim$ (4), z_t is the t-th MFCC vector, $\theta_M = \{\pi_i, \mu_i, \sigma_i | i = 1, 2 \ldots M\}$ is the parameter set for the M-mixture UBM, $p(z_t | \mu_i, \sigma_i)$ is the Gaussian probability density function of the i-th Gaussian mixture component, and γ is a relevance factor [27]. After calculating the statistics using (1) $\sim$ (3), the adapted mean vector μ_i' calculated from (4) is concatenated to form GSV [9], [29], [30]. GSV is therefore denoted by the column vector $\mu_{GSV} = \left[\mu_1'^T \ \mu_2'^T \ \cdots \ \mu_M'^T \right]^T$, which is a "super" vector, whose dimensionality is M times that of μ_i'. If the dimensionality of the MFCC vector is $D \times 1$, then the dimensionality of GSV is $MD \times 1$.

$$\Pr(i | z_t, \theta_M) = \frac{\pi_i \, p(z_t | \mu_i, \sigma_i)}{\sum_{j=1}^{M} \pi_j \, p(z_t | \mu_j, \sigma_j)} \tag{1}$$

$$n_i = \sum_{t=1}^{T} \Pr(i | z_t, \theta_M) \tag{2}$$

$$E_i = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i | z_t, \theta_M) z_t \tag{3}$$

$$\mu_i' = \frac{n_i}{n_i + \gamma} E_i + \frac{\gamma}{n_i + \gamma} \mu_i \tag{4}$$

III. Kernel-Based Projection

In this section, the derivation of the proposed kernel-based projection method is described in detail. Suppose there are N training vectors $\{x_1, x_2 \ldots x_N\}$, for the i-th feature vector x_i (i.e. the raw GSV in this paper), instead of directly using it, x_i is first mapped to another dimensional space using a mapping function $\varphi(x_i)$. The reason of using the mapping is that, in the mapped space, the projection method may find better projecting directions since the feature space is different.

In the mapped space, we would like to find a $D^{(\varphi)} \times P$ projection matrix $V^{(\varphi)}$ where $D^{(\varphi)}$ is the dimensionality of $\varphi(x_i)$ and P is the number of projecting directions, so that after the projection, 1) the feature vectors belonging to the same device are moved closer together, and 2) the feature vectors belonging to different devices are moved farther apart. It is expected that the device information is mainly concentrated in some projecting directions whereas the interfering information is mainly concentrated in other projecting directions, as the projection is oriented to different devices. Through applying this projection, it is hoped that the device information and the interfering information could be well separated into different dimensions, which will be beneficial for recognition. Let y_i be the projected version of $\varphi(x_i)$, then y_i (i.e. the projected GSV in this paper) can be expressed using (5), where $V^{(\varphi)}$ comprises P columns with each column denoting a projecting direction.

$$y_i = V^{(\varphi)^T} \varphi(x_i) \tag{5}$$

The above two goals of the projection method can be achieved by minimizing the objective function in (6) for those pairs of x_i and x_j belonging to the same device, and maximizing the objective function in (6) for those pairs of x_i and x_j belonging to different devices.

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \left\| y_i - y_j \right\|^2 \tag{6}$$

Through utilizing an $N \times N$ coefficient matrix W defined in (7), the two goals of the projection method can be achieved by minimizing the objective function J, which is defined in (8).

$$W_{ij} = \begin{cases} 1 & \text{if } x_i,\ x_j \text{ are from the same device} \\ -1 & \text{if } x_i,\ x_j \text{ are from different devices} \end{cases} \tag{7}$$

$$J = \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \left\| V^{(\varphi)^T}\varphi(x_i) - V^{(\varphi)^T}\varphi(x_j) \right\|^2 \tag{8}$$

To make the projection matrix $V^{(\varphi)}$ unique, the unit-length constraint is applied on each column vector $v_p^{(\varphi)}$ of $V^{(\varphi)}$, as shown in (9) below.

$$v_p^{(\varphi)^T} v_p^{(\varphi)} = 1 \quad for\ p = 1, 2 \ldots P \tag{9}$$

Instead of using the compact expression in (8), J can be expanded in terms of the summation of $v_p^{(\varphi)}$, as shown in (10).

$$\begin{aligned}
J &= \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} (V^{(\varphi)^T}(\varphi(x_i) - \varphi(x_j)))^T \\
&\quad \times (V^{(\varphi)^T}(\varphi(x_i) - \varphi(x_j))) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} (\varphi(x_i) - \varphi(x_j))^T V^{(\varphi)} V^{(\varphi)^T} \\
&\quad \times (\varphi(x_i) - \varphi(x_j)) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} (\varphi(x_i) - \varphi(x_j))^T \\
&\quad \times (\sum_{p=1}^{P} v_p^{(\varphi)} v_p^{(\varphi)^T})(\varphi(x_i) - \varphi(x_j)) \\
&= \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} (\varphi(x_i) - \varphi(x_j))^T v_p^{(\varphi)} v_p^{(\varphi)^T} (\varphi(x_i) - \varphi(x_j)) \\
&= \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_i)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_i) \\
&\quad + \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_j)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_j) \\
&\quad - \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_i)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_j) \\
&\quad - \sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_j)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_i) \\
&= 2\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_i)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_i) \\
&\quad - 2\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} \varphi(x_i)^T v_p^{(\varphi)} v_p^{(\varphi)^T} \varphi(x_j) \\
&= 2\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} v_p^{(\varphi)^T} \varphi(x_i)\varphi(x_i)^T v_p^{(\varphi)} \\
&\quad - 2\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} v_p^{(\varphi)^T} \varphi(x_i)\varphi(x_j)^T v_p^{(\varphi)}
\end{aligned} \tag{10}$$

Through combining all the mapped training vectors into a $D^{(\varphi)} \times N$ matrix $X^{(\varphi)}$ whose i-th column vector is the i-th mapped training vector $\varphi(x_i)$, and introducing an $N \times 1$ vector e whose elements are all one, (10) can be rewritten as (11).

$$\begin{aligned}
J &= 2\sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} (diag(We)) X^{(\varphi)^T} v_p^{(\varphi)} \\
&\quad - 2\sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} W X^{(\varphi)^T} v_p^{(\varphi)} \\
&= 2\sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} (diag(We) - W) X^{(\varphi)^T} v_p^{(\varphi)} \\
&= 2\sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} Z(W) X^{(\varphi)^T} v_p^{(\varphi)}
\end{aligned} \tag{11}$$

where

$$Z(W) = diag(We) - W \tag{12}$$

Combining (9) and (11) and neglecting the constant factor 2 in J, the minimization problem can be formulated as shown in (13), where J' is the new objective function to be minimized in place of J. Minimizing J' is equivalent to minimizing J.

$$\begin{aligned}
&\min\ J' = \sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} Z(W) X^{(\varphi)^T} v_p^{(\varphi)} \\
&subject\ to\ v_p^{(\varphi)^T} v_p^{(\varphi)} = 1 \quad for\ p = 1, 2 \ldots P
\end{aligned} \tag{13}$$

The objective function J' and the constraint can be combined using Lagrange multipliers $\lambda_1, \ldots \lambda_P$, as shown in (14) below, where $L(V^{(\varphi)}, \lambda_1, \ldots \lambda_P)$ is the Lagrangian function to be minimized in place of J'.

$$\begin{aligned}
&L(V^{(\varphi)}, \lambda_1, \cdots \lambda_P) \\
&\quad = \sum_{p=1}^{P} v_p^{(\varphi)^T} X^{(\varphi)} Z(W) X^{(\varphi)^T} v_p^{(\varphi)} - \sum_{p=1}^{P} \lambda_p (v_p^{(\varphi)^T} v_p^{(\varphi)} - 1)
\end{aligned} \tag{14}$$

The optimal solution of (14) can be obtained by setting the partial derivative of $L(V^{(\varphi)}, \lambda_1 \ldots \lambda_P)$ to be zero with respect to $v_p^{(\varphi)}$ and λ_p, as shown in (15).

$$\begin{aligned}
&\frac{\partial L(V^{(\varphi)}, \lambda_1, \cdots \lambda_P)}{\partial v_p^{(\varphi)}} = 2X^{(\varphi)} Z(W) X^{(\varphi)^T} v_p^{(\varphi)} - 2\lambda_p v_p^{(\varphi)} = 0 \\
&\frac{\partial L(V^{(\varphi)}, \lambda_1, \cdots \lambda_P)}{\partial \lambda_p} = v_p^{(\varphi)^T} v_p^{(\varphi)} - 1 = 0 \\
&\qquad\qquad for\ p = 1, 2 \ldots P
\end{aligned} \tag{15}$$

Rearranging the expression in (15), an eigenvalue problem can be formulated as shown in (16). Then $v_p^{(\varphi)}$ is the p-th eigenvector of (16) and λ_p is the p-th eigenvalue. The unit-length constraint can be fulfilled by normalizing the eigenvectors.

$$X^{(\varphi)} Z(W) X^{(\varphi)^T} v_p^{(\varphi)} = \lambda_p v_p^{(\varphi)} \tag{16}$$

Rearranging (16), $v_p^{(\varphi)}$ can be expressed as follows.

$$v_p^{(\varphi)} = X^{(\varphi)} \frac{Z(W) X^{(\varphi)^T} v_p^{(\varphi)}}{\lambda_p} \tag{17}$$

It can be seen from (17) that, in fact $v_p^{(\varphi)}$ can be expressed as a linear combination of the mapped training vectors $\varphi(x_i)$. In other words, if defining the coefficient $c_p^{(\varphi)}$ as in (18), $v_p^{(\varphi)}$ can then be expressed in terms of $X^{(\varphi)}$, as shown in (19).

$$c_p^{(\varphi)} = \frac{Z(W)X^{(\varphi)^T} v_p^{(\varphi)}}{\lambda_p} \tag{18}$$

$$v_p^{(\varphi)} = X^{(\varphi)} c_p^{(\varphi)} \tag{19}$$

By substituting $v_p^{(\varphi)}$ in (16) by (19), (16) can then be rewritten as (20).

$$X^{(\varphi)} Z(W) X^{(\varphi)^T} X^{(\varphi)} c_p^{(\varphi)} = \lambda_p X^{(\varphi)} c_p^{(\varphi)} \tag{20}$$

By multiplying $X^{(\varphi)^T}$ on both sides of (20), we have,

$$X^{(\varphi)^T} X^{(\varphi)} Z(W) X^{(\varphi)^T} X^{(\varphi)} c_p^{(\varphi)} = \lambda_p X^{(\varphi)^T} X^{(\varphi)} c_p^{(\varphi)} \tag{21}$$

By substituting $X^{(\varphi)^T} X^{(\varphi)}$ in (21) by an $N \times N$ kernel matrix K, whose ij-th entry is given by a kernel function $k(x_i,x_j)$ defined in (22) below, (21) can be reformulated as (23).

$$\left(X^{(\varphi)^T} X^{(\varphi)}\right)_{ij} = K_{ij} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \tag{22}$$

$$K Z(W) K c_p^{(\varphi)} = \lambda_p K c_p^{(\varphi)} \tag{23}$$

By introducing a diagonal matrix $\Lambda = diag(\lambda_1, \ldots \lambda_P)$ whose size is $P \times P$, a compact form of (16) can be obtained, which is given by (24). By defining an $N \times P$ coefficient matrix $C^{(\varphi)}$ whose p-th column vector is $c_p^{(\varphi)}$, a compact form of (23) can be obtained, which is given by (25).

$$X^{(\varphi)} Z(W) X^{(\varphi)^T} V^{(\varphi)} = V^{(\varphi)} \Lambda \tag{24}$$

$$K Z(W) K C^{(\varphi)} = K C^{(\varphi)} \Lambda \tag{25}$$

Now instead of directly solving (24) for $v_p^{(\varphi)}$, it is feasible to first solve (25) for $c_p^{(\varphi)}$ and then make use of (19) to solve $v_p^{(\varphi)}$. As can be seen from (25), the sizes of $KZ(W)K$ and K are both $N \times N$, meaning that there are at most N independent $c_p^{(\varphi)}$, namely $P \leq N$. At the very beginning, the raw input feature vector x_i is first mapped to a new vector $\varphi(x_i)$. On using (24) to find the projection matrix $V^{(\varphi)}$, the mapping $\varphi(x_i)$ has to be expressed explicitly. However, on using (25) to find the projection matrix, it is only necessary to know the inner product of the two mapped vectors $\varphi(x_i)^T \varphi(x_j)$, instead of knowing the mapping function. This kernel trick gives great flexibility, since it is possible to employ any valid kernel function without knowing the explicit mapping, as long as the kernel function satisfies the Mercer's condition [31]. In this paper, the Gaussian kernel defined in (26) is employed, where d is the kernel parameter.

$$k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) = e^{-\|x_i - x_j\|^2 / d} \tag{26}$$

On using the Gaussian kernel in (26), $\varphi(x_i)$ and $\varphi(x_j)$ are of infinite dimensionality if written out explicitly (i.e. $D^{(\varphi)}$ has infinite dimensions), but fortunately, on using the kernel version (25), it is only necessary to calculate the inner product $\varphi(x_i)^T \varphi(x_j)$, which is finite [31]. Gaussian kernel has the capability to map the feature vector onto an infinite dimensional space. In this mapped space, as the dimensionality is higher than that of the original feature space, the projected feature vectors may work better. This kind of mapping can only be applicable when using (25) to find the projection matrix. It is impossible to use (24) to find the projection matrix, as the column vectors of $X^{(\varphi)}$ have infinite dimensions. In addition, on using the Gaussian kernel, as $D^{(\varphi)} >> N$, K will be of full rank (assuming the training vectors are linearly independent of each other), then solving (25) is equivalent to finding the eigenvectors of $Z(W)KC^{(\varphi)} = C^{(\varphi)}\Lambda$.

After solving (25) and obtaining $c_p^{(\varphi)}$, for a given vector t, its projected version t' can be calculated using (27), where t'_p is the p-th element of t', and $(c_p^{(\varphi)})_i$ is the i-th element of $c_p^{(\varphi)}$.

$$\begin{aligned} t'_p &= v_p^{(\varphi)^T} t = (X^{(\varphi)} c_p^{(\varphi)})^T t = c_p^{(\varphi)^T} X^{(\varphi)^T} t \\ &= \sum_{i=1}^{N} (c_p^{(\varphi)})_i k(x_i, t) \end{aligned} \tag{27}$$

After finding t'_p, $v_p^{(\varphi)}$ can be normalized implicitly by normalizing t'_p with respect to $v_p^{(\varphi)}$ as shown in (28).

$$\begin{aligned} \frac{t'_p}{\left\| v_p^{(\varphi)} \right\|} &= \frac{v_p^{(\varphi)^T} t}{\sqrt{v_p^{(\varphi)^T} v_p^{(\varphi)}}} = \frac{c_p^{(\varphi)^T} X^{(\varphi)^T} t}{\sqrt{c_p^{(\varphi)^T} X^{(\varphi)^T} X^{(\varphi)} c_p^{(\varphi)}}} \\ &= \frac{\sum_{i=1}^{N} (c_p^{(\varphi)})_i k(x_i, t)}{\sqrt{c_p^{(\varphi)^T} K c_p^{(\varphi)}}} \end{aligned} \tag{28}$$

IV. Sparse Representation and SRC

In order to apply the Sparse Representation based Classifier (SRC) for a given feature vector, the sparse representation of the feature vector must be computed first. Suppose there is a matrix A whose dimensionality is $D \times N$, (this matrix A is often called the dictionary), for a given input vector a whose dimensionality is $D \times 1$, the sparse representation of a is obtained by solving the optimization problem defined in (29) below, where a_s is the sparse representation of a, and both a_s and b are of dimensionality $N \times 1$. $||.||_0$ is L0 norm.

$$a_s = \arg\min_b \|b\|_0 \quad subject\ to\ a = Ab \tag{29}$$

The optimization problem in (29) aims at finding a linear combination of the column vectors of A, such that the number of nonzero coefficients are minimized, i.e. the number of nonzero elements in a_s is minimal [32]. Under some conditions, the solution of (29) can be approximated by solving the optimization problem in (30). If the solution is sparse enough, (29) is equivalent to (30), where (30) can be more easily solved because L0 norm is replaced by L1 norm [32].

$$a_s = \arg\min_b \|b\|_1 \quad subject\ to\ a = Ab \tag{30}$$

Having the sparse representation, the Sparse Representation based Classifier (SRC) was introduced in [33]. SRC is found to be good in telephone handset recognition [6]–[8] and mobile phone verification [11] and comparison [12]. According to [33], for a given input feature vector (i.e. the raw GSV or

TABLE I
MICROPHONE DATASET

Set	Microphone model	Number of speeches		Duration
		Training	Testing	
M1	AKG C410B Head Mounted	240	260	2s ~ 5s
M2	AKH D80S Desktop	240	260	
M3	SONY ECM 66B Lapel	240	260	
M4	TARGET Lapel	240	260	
UBM	All the models	599		10s ~ 100s

the projected GSV), the corresponding sparse representation is obtained by solving (30). In this paper, the optimization problem in (30) is solved by the Basis Pursuit (BP) algorithm [34] implemented by SparseLab [35]. The i-th column vector A_i of the dictionary A in (30) is the i-th raw GSV or the i-th projected GSV in the training set with L2 normalization. In other words, the training data are used to form the dictionary. SRC then works as follows.

Suppose there are totally K classes. After obtaining the sparse representation a_s for the feature vector a, the residual $r^{(k)}(a)$ of a with respect to class k is calculated using (31), where $a_s^{(k)}$ is an $N \times 1$ vector whose i-th element is given by (32).

$$r^{(k)}(a) = \left\| a - Aa_s^{(k)} \right\|_2 \quad for\ k = 1, 2 \ldots K \tag{31}$$

where

$$(a_s^{(k)})_i = \begin{cases} (a_s)_i & if\ A_i\ belongs\ to\ class\ k \\ 0 & otherwise \end{cases} \tag{32}$$

Having obtained the residual $r^{(k)}(a)$, the feature vector a will then be classified to the class having the minimum residual, as given by (33).

$$class(a) = \arg\min_k r^{(k)}(a) \quad for\ k = 1, 2 \ldots K \tag{33}$$

V. MICROPHONE SPEECH DATASET

In this paper, Ahumada-25 is used to carry out the microphone recognition task. Ahumada-25 is a part of AHUMADA Spanish speech corpus [36]. It consists of the speech recordings coming from 25 speakers. The speeches are recorded using 4 different microphones, as listed in Table I. The contents of the speeches vary from isolated numbers to texts, and from sentences to continuous speeches. This dataset is then divided into three separate subsets: a training set, a testing set, and a UBM set. The training set consists of the speeches coming from 12 speakers, while the testing set consists of the speeches coming from the remaining 13 speakers. There are 20 speech recordings coming from each speaker used in the training set and the testing set. Another 599 speech recordings are used to construct the UBM set, where all the 25 speakers are involved, and each speaker contributes almost the same number of speech recordings. Totally 960 speech recordings are used in the training set, 1040 speech recordings are used in the testing set, and 599 speech recordings are used in the UBM set.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In Part A of this section, different methods used to form the feature vector are compared, including the averaged frame-level features and GSV, as illustrated in Fig. 1. Different UBMs are used to calculate GSV, in order to investigate how the performance of GSV will be influenced by the number of mixture components in the UBM. The number of mixture components in the UBM determines the dimensionality of GSV, and the larger the number of mixture components, the higher the dimensionality will be. Different relevance factors are also used in the calculation of GSV. Two different kinds of classifiers, namely linear SVM and SRC, are employed. The linear SVM is implemented using LIBSVM [37]. In Part B and C of this section, the performance of the raw GSV and the projected GSV is compared, employing both linear SVM and SRC as the classifier. On using the projection method, different values of the kernel parameter are evaluated. The value of the kernel parameter is chosen in a heuristic way, varying from 50 (smaller than the dimensionality of the raw GSV) to 2000 (larger than or approximately the same as the dimensionality of the raw GSV), so as to investigate the performance change trend with respect to the kernel parameter. The number of projecting directions is set to be equal to the number of training vectors, namely $P = N$, hoping that making use of all the projecting directions may give the best performance. In Part D of this section, how different UBMs influence the effectiveness of the projection method is investigated, employing both linear SVM and SRC as the classifier. In Part E of this section, whether the performance improvement offered by the projection method is statistically significant, is illustrated with respect to different confidence levels. In Part F of this section, a brief summary is given.

A. Investigation of the Influence of Different UBMs on GSV

In this part, the performance of the averaged frame-level feature and GSV is compared, and how the performance of GSV will be influenced by the number of mixture components in the UBM is investigated. The recognition results of using the averaged frame-level feature and GSV are shown in Table II and Table III. Table II shows the results on employing SVM as the classifier. Table III shows the results on employing SRC as the classifier. On using GSV, different relevance factors (γ = 5, 10, 15, 20) and different UBMs (with M = 32, 64, 128, 256) are evaluated. As explained in Section II, the larger the M, the higher the dimensionality of GSV. The results are also illustrated in Fig. 2.

Regarding the feature vector, GSV can outperform the averaged MFCC when a large enough number of mixture components in the UBM (i.e. M = 64, 128, 256) is used. The reasons could be explained from two aspects. First, GSV is of high dimensionality and makes full use of each frame-level feature; while the averaged MFCC simply averages all the frame-level features, which results in loss of information. So intrinsically, GSV can carry more information than the averaged MFCC. Second, GSV is calculated based on a UBM, and therefore can absorb extra information from the UBM.

TABLE II

MICROPHONE RECOGNITION ACCURACY USING AVERAGED MFCC AND GSV EMPLOYING SVM AS CLASSIFIER (%)

Feature vector	No. of mixtures M in UBM	Relevance factor γ	Recognition accuracy
Averaged MFCC	n/a	n/a	79.23
Raw GSV	32	5	78.17
		10	78.65
		15	79.62
		20	79.71
	64	5	83.27
		10	84.71
		15	85.19
		20	85.48
	128	5	85.58
		10	85.67
		15	85.10
		20	84.81
	256	5	85.10
		10	84.04
		15	83.27
		20	82.98

TABLE III

MICROPHONE RECOGNITION ACCURACY USING AVERAGED MFCC AND GSV EMPLOYING SRC AS CLASSIFIER (%)

Feature vector	No. of mixtures M in UBM	Relevance factor γ	Recognition accuracy
Averaged MFCC	n/a	n/a	75.87
Raw GSV	32	5	69.04
		10	69.62
		15	71.15
		20	70.58
	64	5	77.02
		10	79.04
		15	78.56
		20	78.65
	128	5	85.77
		10	84.13
		15	83.46
		20	82.69
	256	5	85.67
		10	85.00
		15	83.46
		20	83.08

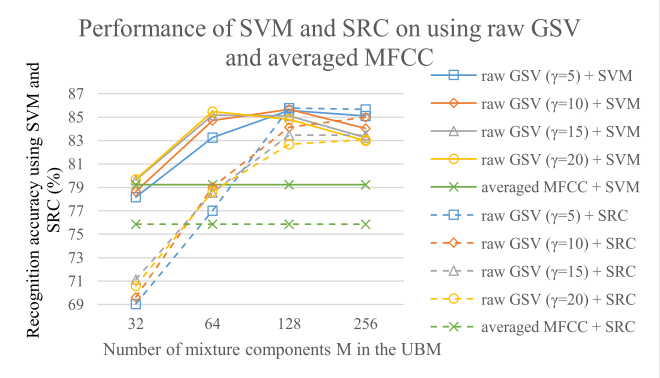


Fig. 2. Microphone recognition accuracy using raw GSV and the averaged MFCC as the feature vector, and SVM and SRC as the classifier.

It is also noticed that increasing M may not always improve the performance of GSV. The performance of GSV tends to stabilize when M is large; for example, when $M = 128$.

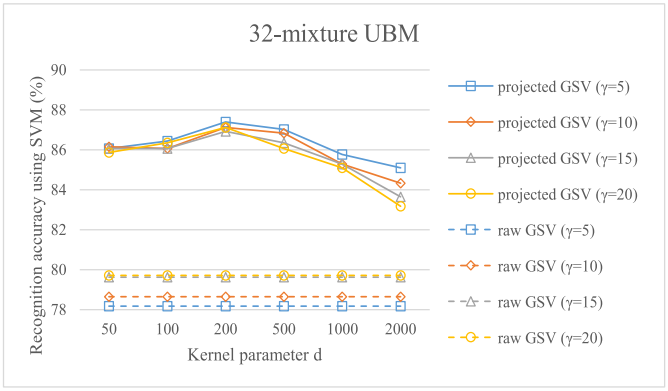


Fig. 3. Microphone recognition accuracy employing SVM as the classifier, with the GSV adapted from a 32-mixture UBM.

This may be explained from the construction procedure of the UBM, which involves lots of frame-level features (i.e. MFCC vectors in this paper). These frame-level features carry both the source speech information and the recording device information. When people are listening to speeches recorded using different devices, usually they are unable to distinguish one recording device from another because the effect due to device response is weaker than the source speech signal. This causes the UBM to model more of the source speech signal than the device response. Hence, in the UBM, the device response information is distorted. This distortion becomes severer as the value of M increases. This results in the quality of GSV being affected.

Regarding the classifier, on using the averaged MFCC, SVM outperforms SRC. On using GSV, SVM can perform better than SRC when M is small (e.g. $M = 32, 64$), but may perform worse than SRC when M is large (e.g. $M = 256$). The different behaviors of SVM and SRC are probably owing to the different classification mechanisms adopted by SVM and SRC. SVM builds a model for classification and therefore the performance is more dependent on the quality of the feature vector (i.e. GSV). On the contrary, SRC does not build any model, and the classification is based on the reconstruction error of a group of feature vectors, as can be seen from (31) and (32). Hence, SRC is more resilient and less dependent on the quality of the feature vector (i.e. GSV). However, practically, SVM is faster than SRC.

B. Effectiveness of the Projection Method With SVM as the Classifier

In this part, SVM is employed as the classifier for the comparison of the raw GSV and the projected GSV in doing microphone recognition. The results are shown in Table IV; the effectiveness of the projection method on GSV calculated from different UBMs (i.e. number of mixture components $M = 32$, 64, 128) are also illustrated in Figs. 3 $\sim$ 5. On using the projection method, different kernel parameters d are evaluated.

From Figs. 3 $\sim$ 5, it can be seen that the projected GSV with a suitable choice of the kernel parameter can give improvement over the raw GSV, which exhibits the effectiveness of the kernel-based projection method. Comparing Figs. 3 $\sim$ 5,

TABLE IV

MICROPHONE RECOGNITION ACCURACY USING GSV AND PROJECTED GSV EMPLOYING SVM AS CLASSIFIER (%)

No. of mixtures in UBM	Feature vector	Kernel parameter d	Relevance factor γ 5	10	15	20
32	Raw GSV	n/a	78.17	78.65	79.62	79.71
	Projected GSV	50	86.06	86.15	86.06	85.87
		100	86.44	86.06	86.06	86.35
		200	87.40	87.12	86.92	87.12
		500	87.02	86.83	86.35	86.06
		1000	85.77	85.29	85.29	85.10
		2000	85.10	84.33	83.65	83.17
64	Raw GSV	n/a	83.27	84.71	85.19	85.48
	Projected GSV	50	88.27	88.46	88.37	87.88
		100	89.04	88.65	88.27	87.79
		200	88.46	88.37	88.94	88.17
		500	89.04	88.46	87.31	86.63
		1000	87.88	87.12	86.25	85.58
		2000	86.83	86.15	84.71	82.88
128	Raw GSV	n/a	85.58	85.67	85.10	84.81
	Projected GSV	50	88.08	87.40	86.15	85.10
		100	88.08	86.54	85.77	85.19
		200	87.12	86.73	85.87	85.77
		500	87.79	87.21	86.83	85.77
		1000	87.88	86.73	85.48	82.98
		2000	87.50	85.58	81.35	80.87

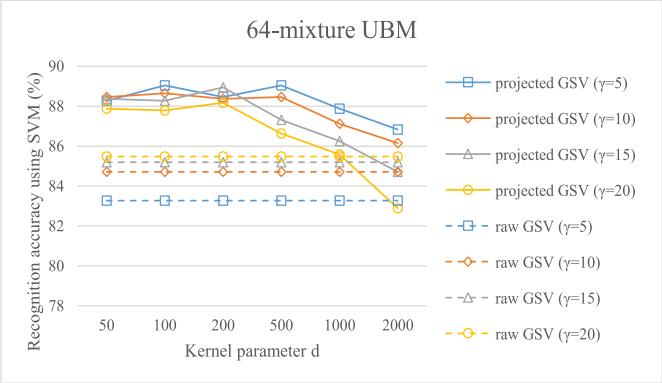


Fig. 4. Microphone recognition accuracy employing SVM as the classifier, with the GSV adapted from a 64-mixture UBM.

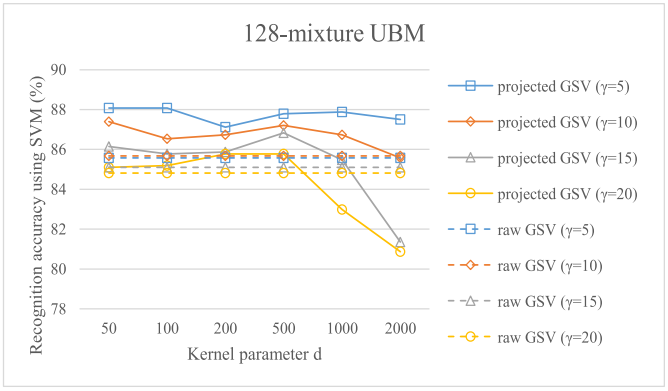


Fig. 5. Microphone recognition accuracy employing SVM as the classifier, with the GSV adapted from a 128-mixture UBM.

TABLE V

MICROPHONE RECOGNITION ACCURACY USING GSV AND PROJECTED GSV EMPLOYING SRC AS CLASSIFIER (%)

No. of mixtures in UBM	Feature vector	Kernel parameter d	Relevance factor γ 5	10	15	20
32	Raw GSV	n/a	69.04	69.62	71.15	70.58
	Projected GSV	50	81.35	82.50	84.23	84.52
		100	83.94	84.52	85.38	85.77
		200	85.48	86.06	86.63	86.63
		500	86.15	86.35	86.25	85.77
		1000	86.54	85.87	84.81	84.23
		2000	86.15	84.42	83.37	82.79
64	Raw GSV	n/a	77.02	79.04	78.56	78.65
	Projected GSV	50	82.69	84.81	86.92	87.88
		100	84.81	87.40	87.98	87.88
		200	86.73	87.69	87.79	87.50
		500	86.63	86.92	87.12	86.92
		1000	85.67	85.77	85.58	85.58
		2000	84.62	84.13	83.94	83.65
128	Raw GSV	n/a	85.77	84.13	83.46	82.69
	Projected GSV	50	84.04	86.73	87.69	87.98
		100	87.21	88.17	87.60	87.69
		200	88.65	88.27	87.60	87.60
		500	88.37	87.50	86.63	85.48
		1000	87.88	86.44	85.38	84.23
		2000	87.40	85.67	84.33	83.65

it seems the larger the M, the less heavily the solid polylines will overlap. This indicates that the relevance factor γ increases its influence on the projected GSV with the increase of M. In addition, it seems the larger the M, the smaller the improvement of the projected GSV over the raw GSV. This is probably owing to two reasons. First, the larger the M, the possibly better the performance of the raw GSV, due to the increase of the dimensionality, as illustrated in Fig. 2. So, the performance of the raw GSV may be improved by increasing M and therefore the performance gap between the raw GSV and the projected GSV may be narrowed. Second, the larger the M, the possibly lower the quality of the GSV, due to the quality of the UBM, as explained in Part A of this section. So, the effectiveness of the projection method may be affected, resulting in the performance gap between the raw GSV and the projected GSV being narrowed.

C. Effectiveness of the Projection Method With SRC as the Classifier

In this part, SRC is employed as the classifier for the comparison of the raw GSV and the projected GSV in doing microphone recognition. The results are shown in Table V; the effectiveness of the projection method on GSV calculated from different UBMs are also illustrated in Figs. 6 ∼ 8. On using the projection method, different kernel parameters d are evaluated.

It can be seen from Figs. 6 ∼ 8, on employing SRC as the classifier with suitably chosen kernel parameters, the projected GSV can also give improvement over the raw GSV, which is similar to what has been observed on employing SVM as the classifier. Like the observation in Part B of this section, the larger the M, the more influence the relevance factor will exert on the projected GSV, and the smaller improvement the projection method will give.

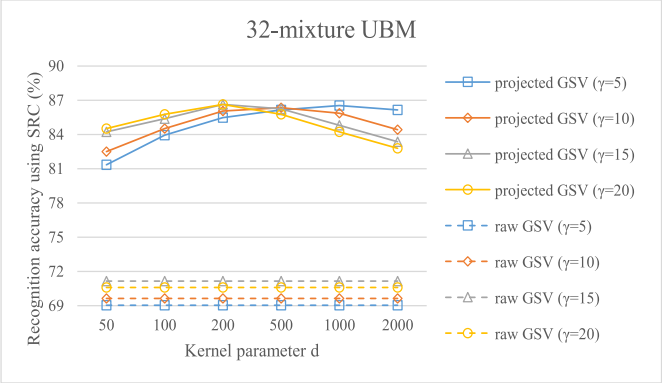


Fig. 6. Microphone recognition accuracy employing SRC as the classifier, with the GSV adapted from a 32-mixture UBM.

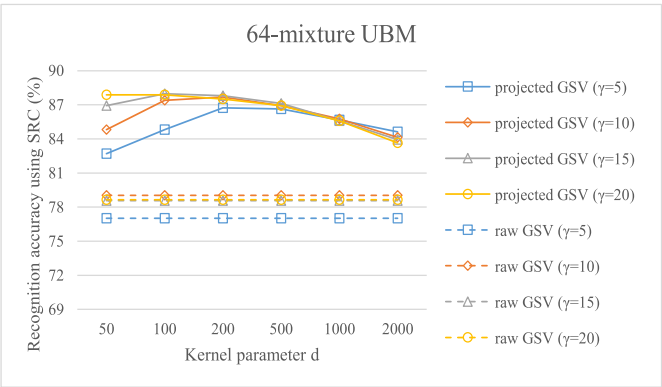


Fig. 7. Microphone recognition accuracy employing SRC as the classifier, with the GSV adapted from a 64-mixture UBM.

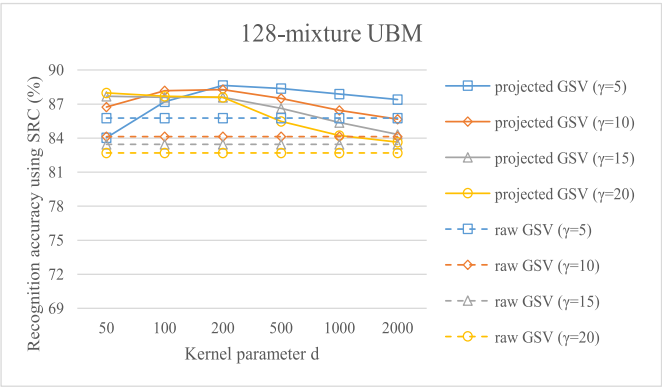


Fig. 8. Microphone recognition accuracy employing SRC as the classifier, with the GSV adapted from a 128-mixture UBM.

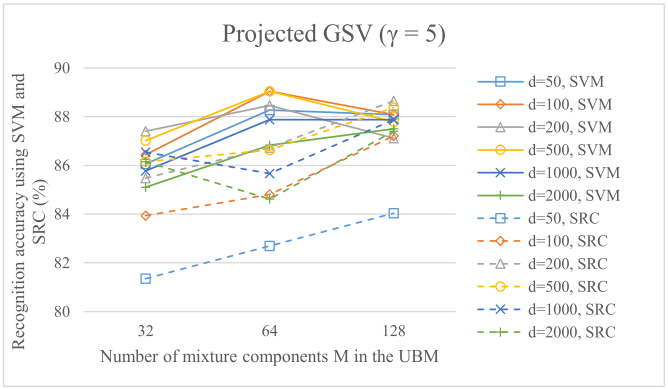


Fig. 9. Microphone recognition accuracy using projected GSV, with the relevance factor equal 5.

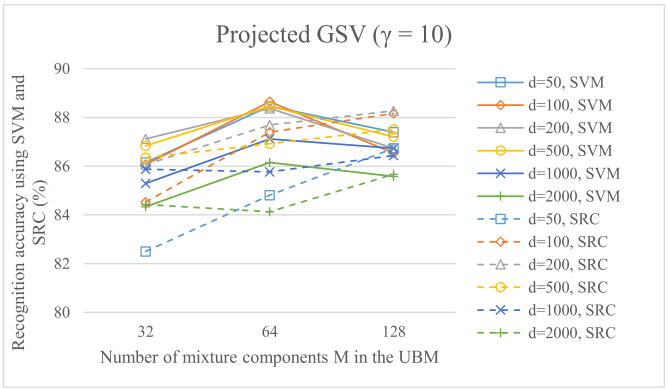


Fig. 10. Microphone recognition accuracy using projected GSV, with the relevance factor equal 10.

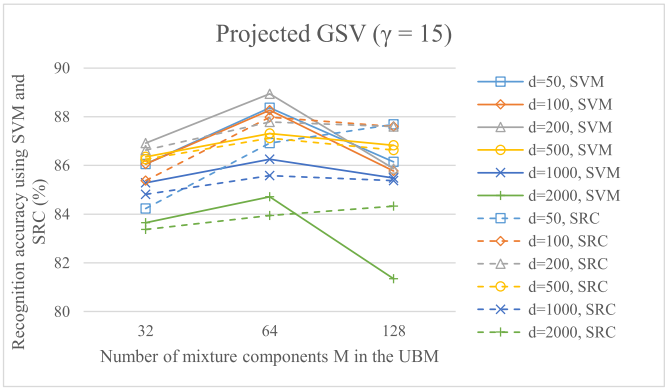


Fig. 11. Microphone recognition accuracy using projected GSV, with the relevance factor equal 15.

D. Investigation of the Influence of Different UBMs on the Projection Method

In this part, the influence of the number of mixture components on the effectiveness of the projection method is investigated. The recognition results of the projected GSV are shown in Figs. 9 ∼ 12, where each figure corresponds to one relevance factor.

From Figs. 9 ∼ 12, on using SVM, the performance of the projected GSV starts to degrade when M is large (e.g. comparing $M = 64$ and $M = 128$). On using SRC, the performance of the projected GSV tends to stabilize when M is large (e.g. comparing $M = 64$ and $M = 128$ in Fig. 12). The different behaviors between SVM and SRC are induced by the different classification mechanisms adopted by SVM and SRC as explained in Part A of this section. SVM is a model-based classifier, while SRC is an example-based classifier. These figures also indicate that too many mixture components may lower the quality of GSV, and consequently may not benefit the projection method much.

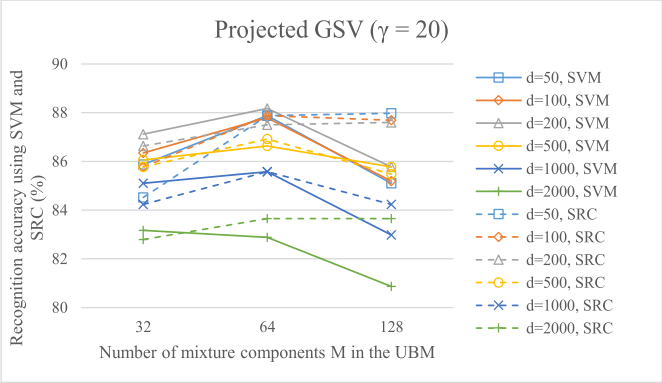


Fig. 12. Microphone recognition accuracy using projected GSV, with the relevance factor equal 20.

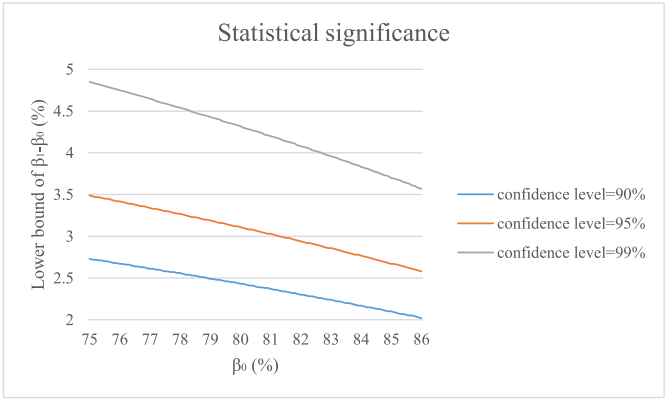


Fig. 13. Lower bound for the improvement in recognition accuracy to be statistically significant.

Comparing SVM and SRC, in general, SVM outperforms SRC when M is small (e.g. $M = 32, 64$), whereas SRC outperforms SVM when M is large (e.g. $M = 128$). Besides, on using SVM, the performance of the projected GSV degrades faster (e.g. M increases from 64 to 128) when the relevance factor is large (e.g. comparing $\gamma = 5$ and $\gamma = 10$). On using SRC, the performance of the projected GSV tends to stabilize faster (e.g. M increases from 64 to 128) when the relevance factor is large (e.g. comparing $\gamma = 15$ and $\gamma = 20$). As can be seen from (4), for a speech recording, its corresponding GSV contains the information from this recording as well as the UBM. The larger the relevance factor, the more information GSV obtains from the UBM. Since the UBM is shared by all GSVs, the more information GSV obtains from the UBM, the more similar to the others this GSV will be. In other words, the larger the relevance factor, the higher the similarity of different GSVs. Since the projection method aims to group those GSVs coming from the same device and separate those GSVs coming from different devices, the similarity of different GSVs indeed affects the effectiveness of the projection method.

E. Statistical Significance of the Improvement Offered by the Projection Method

In this part, whether the performance improvement offered by the projection method is statistically significant, is evaluated with respect to different confidence levels. Let $(1-\alpha)$ be a confidence level where $0 \leq \alpha \leq 1$, z_α be a value related to α, β_0 be the recognition accuracy of using the raw GSV and β_1 be the recognition accuracy of using the projected GSV. According to [38], if the relationship between β_0 and β_1 satisfies the inequality as given by (34) where N is the number of testing data ($N = 1040$ in this paper), then we can say that the performance improvement is statistically significant with the confidence level being $(1-\alpha)$.

$$\beta_1 - \beta_0 \geq \frac{z_\alpha}{\sqrt{N}}\sqrt{(1-\beta_0)+(1-\beta_1)} \tag{34}$$

Taking square on both sides of (34), an equivalent inequality can be obtained as given by (35), where a new variable Δ is defined in (36) for simplification.

$$\begin{aligned}
&(\beta_1-\beta_0)^2 \geq \frac{z_\alpha^2}{N}(2-\beta_0-\beta_1)\\
&\Leftrightarrow\\
&\beta_1^2 - 2\beta_0\beta_1 + \beta_0^2 \geq \frac{2z_\alpha^2}{N} - \frac{z_\alpha^2}{N}\beta_0 - \frac{z_\alpha^2}{N}\beta_1\\
&\Leftrightarrow\\
&\beta_1^2 + (\frac{z_\alpha^2}{N} - 2\beta_0)\beta_1 + (\beta_0^2 + \frac{z_\alpha^2}{N}\beta_0 - \frac{2z_\alpha^2}{N}) \geq 0\\
&\Leftrightarrow\\
&\left(\beta_1 - \frac{-(\frac{z_\alpha^2}{N}-2\beta_0)-\sqrt{\Delta}}{2}\right)\left(\beta_1 - \frac{-(\frac{z_\alpha^2}{N}-2\beta_0)+\sqrt{\Delta}}{2}\right) \geq 0
\end{aligned} \tag{35}$$

where

$$\Delta = (\frac{z_\alpha^2}{N} - 2\beta_0)^2 - 4(\beta_0^2 + \frac{z_\alpha^2}{N}\beta_0 - \frac{2z_\alpha^2}{N}) \tag{36}$$

According to (35), if $\beta_1 \geq \left(-(z_\alpha^2/N - 2\beta_0) + \sqrt{\Delta}\right)/2$, then the difference between β_0 and β_1 is considered to be statistically significant with the confidence level being $(1-\alpha)$. $\left(-(z_\alpha^2/N - 2\beta_0) + \sqrt{\Delta}\right)/2$ is namely the lower bound for the recognition accuracy to be statistically significant. With this lower bound, it is then feasible to evaluate whether the performance of the projected GSV is statistically significantly better than that of the raw GSV.

Fig. 13 shows the lower bound for the performance improvement to be statistically significant with respect to different confidence levels. According to [38], $z_\alpha = 2.33$ corresponds to the confidence level of 99% (i.e. $\alpha = 0.01$), $z_\alpha = 1.65$ corresponds to the confidence level of 95% (i.e. $\alpha = 0.05$), $z_\alpha = 1.28$ corresponds to the confidence level of 90% (i.e. $\alpha = 0.1$). Fig. 14 and Fig. 15 illustrate the recognition accuracy of using the projected GSV and the raw GSV, and the lower bound for the recognition accuracy of the projected GSV to be statistically significantly better than that of the raw GSV (dotted polylines). The relevance factor γ is chosen to be 5 and the kernel parameter d is chosen to be 200 as an example. On using SVM as the classifier (Fig. 14), when the number of mixture components in the UBM is small

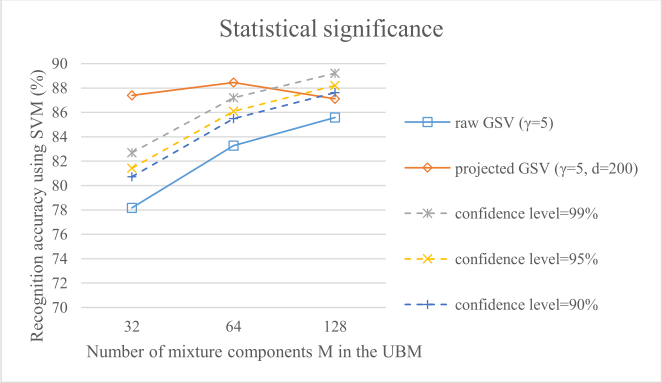


Fig. 14. Statistical significance of the performance improvement offered by the projection method employing SVM as the classifier.

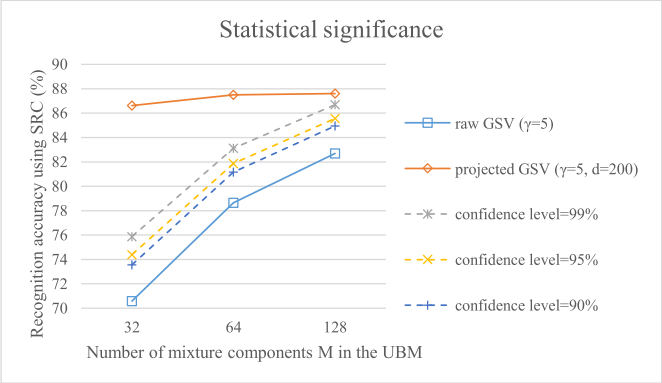


Fig. 15. Statistical significance of the performance improvement offered by the projection method employing SRC as the classifier.

(i.e. $M = 32, 64$), the performance improvement offered by the projection method is statistically significant with the confidence level being 99%, while the number of mixture components is large (i.e. $M = 128$), the performance improvement is not statistically significant or at least the confidence level is below 90%. On using SRC as the classifier (Fig. 15), the performance improvement offered by the projection method is statistically significant with the confidence level being 99%, for $M = 32$, 64 and 128.

F. A Brief Summary

By employing both SVM and SRC as the classifier, it is shown that the kernel-based projection method can improve the performance of GSV, when suitable kernel parameters are used. On increasing the number of mixture components in the UBM, the performance of the raw GSV and the projected GSV cannot always be improved, and the performance tends to stabilize when the number of mixture components in the UBM is large. As explained in previous parts, although the increase in the number of mixture components in the UBM increases the dimensionality of GSV and consequently increases the information that GSV can provide to the classifier (i.e. SVM or SRC in this paper), this increase may distort the device information embedded in GSV and consequently lower the quality of GSV.

By comparing the recognition results using SVM and SRC, it has been observed that when the number of mixture components in the UBM is small (i.e. the dimensionality of GSV is small), SVM is more effective; when the number of mixture components in the UBM is large, SRC tends to work better. The difference is caused by the different classification mechanisms adopted by SVM and SRC. SVM is a model-based classifier while SRC is an example-based classifier. Thus, SVM is more dependent on the quality of the feature vector (i.e. GSV). Since SRC relies on a group of feature vectors for classification, it is less dependent on the quality of the feature vector. However, practically SVM is usually faster than SRC in doing classification.

VII. Conclusion

In this paper, the focus is on a closed-set microphone recognition task. In terms of feature extraction, different feature formation methods have been compared, including the averaged frame-level feature and Gaussian Supervector (GSV). It is shown that GSV can outperform the averaged frame-level feature, as GSV makes full use of all the frame-level features. The influence of the Universal Background Model (UBM) on GSV has also been investigated. It is found that, increasing the number of mixture components in the UBM (i.e. increasing the dimensionality of GSV) may not always improve the performance, because the device information carried by GSV may be more severely distorted on increasing the dimensionality. In terms of recognition, the performance of using linear Support Vector Machine (SVM) and Sparse Representation based Classifier (SRC) as the classifier, has been compared. Although SVM and SRC exhibit different behaviours in different situations (e.g. different dimensionalities of GSV), the performances of these two classifiers are basically quite similar.

Facing the situation that the raw GSV may not always perform very well, a kernel-based projection method is proposed, which can project the original feature vector (i.e. GSV in this paper) onto another dimensional space. As GSV embeds both the speech information (which is useless) as well as the device information (which is useful), hopefully the proposed projection method can separate these two types of information into different dimensions in the projected GSV, which benefits the recognition. Experimental results demonstrate that, with suitably chosen kernel parameters, the projected GSV can outperform the raw GSV, no matter using SVM or SRC as the classifier. The improvement shows the effectiveness and the potential of the kernel-based projection method in microphone recognition.

References

[1] S. Gupta, S. Cho, and C.-C. J. Kuo, "Current developments and future trends in audio authentication," *IEEE Multimedia Mag.*, vol. 19, no. 1, pp. 50–59, Jan. 2012.

[2] C. Kraetzer, A. Oermann, J. Dittmann, and A. Lang, "Digital audio forensics: A first practical evaluation on microphone and environment classification," in *Proc. 9th Workshop Multimedia Secur.*, Dallas, TX, USA, Sep. 2007, pp. 63–74.

[3] C. Kraetzer, M. Schott, and J. Dittmann, "Unweighted fusion in microphone forensics using a decision tree and linear logistic regression models," in *Proc. 11th ACM Workshop Multimedia Secur.*, Sep. 2009, pp. 49–56.

[4] Ö. Eskidere, "Source microphone identification from speech recordings based on a Gaussian mixture model," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 754–767, Apr. 2014.

[5] O. Eskidere and A. Karatutlu, "Source microphone identification using multitaper MFCC features," in *Proc. 9th Int. Conf. Electr. Electron. Eng. (ELECO)*, Nov. 2015, pp. 227–231.

[6] Y. Panagakis and C. Kotropoulos, "Automatic telephone handset identification by sparse representation of random spectral features," in *Proc. Multimedia Secur.*, Sep. 2012, pp. 91–96.

[7] C. Kotropoulos, "Telephone handset identification using sparse representations of spectral feature sketches," in *Proc. Int. Workshop Biometrics Forensics (IWBF)*, Lisbon, Portugal, Apr. 2013, pp. 1–4.

[8] Y. Panagakis and C. Kotropoulos, "Telephone handset identification by feature selection and sparse representations," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 73–78.

[9] D. Garcia-Romero and C. Y. Espy-Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 1806–1809.

[10] C. L. Kotropoulos, "Source phone identification using sketches of features," *IET Biometric*, vol. 3, no. 2, pp. 75–83, Jun. 2014.

[11] L. Zou, Q. He, and X. Feng, "Cell phone verification from speech recordings using sparse representation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 1787–1791.

[12] L. Zou, Q. He, J. Yang, and Y. Li,, "Source cell phone matching from speech recordings by sparse representation and KISS metric," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2079–2083.

[13] H. Q. Vu, S. Liu, X. Yang, Z. Li, and Y. Ren, "Identifying microphone from noisy recordings by using representative instance one class-classification approach," *J. Netw.*, vol. 7, no. 6, pp. 908–917, 2012.

[14] L. Cuccovillo and P. Aichroth, "Open-set microphone classification via blind channel analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2074–2078.

[15] F. Kurniawan, M. S. M. Rahim, M. S. Khalil, and M. K. Khan, "Statistical based audio forensic on identical microphones," *Int. J. Elect. Comput. Eng.*, vol. 6, no. 5, pp. 2211–2218, 2016.

[16] H. Malik and J. W. Miller, "Microphone identification using higher-order statistics," in *Proc. 46th Int. Conf., Audio Forensics*, Denver, MI, USA, Jun. 2012, pp. 1–10.

[17] L. Cuccovillo, S. Mann, M. Tagliasacchi, and P. Aichroth, "Audio tampering detection via microphone classification," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep./Oct. 2013, pp. 177–182.

[18] C. Hanilçi, F. Ertaş, T. Ertaş, and Ö. Eskidere, "Recognition of brand and models of cell-phones from recorded speech signals," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 625–634, Apr. 2012.

[19] R. Buchholz, C. Kraetzer, and J. Dittmann, "Microphone classification using Fourier coefficients," in *Information Hiding* (Lecture Notes in Computer Science), vol. 5806. Berlin, Germany: Springer, 2009, pp. 235–246.

[20] C. Hanilçi and T. Kinnunen, "Source cell-phone recognition from recorded speech using non-speech segments," *Digit. Signal Process.*, vol. 35, pp. 75–85, Dec. 2014.

[21] R. Aggarwal, S. Singh, A. K. Roul, and N. Khanna,, "Cellphone identification using noise estimates from recorded audio," in *Proc. Int. Conf. Commun. Signal Process.*, Melmaruvathur, India, Apr. 2014, pp. 1218–1222.

[22] Y. Jiang and F. H. F. Leung, "Mobile phone identification from speech recordings using weighted support vector machine," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 963–968.

[23] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2005, pp. 629–632.

[24] A. Solomonoff, W. M. Campbell, and C. Quillen, "Nuisance attribute projection," in *Speech Communication*. Amsterdam, The Netherlands: Elsevier Science BV, 2007, pp. 1–73.

[25] X. Huang, A. Acero, and H.-W. Hon, "Speech Signal Representations," in *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, vol. 6. Upper Saddle River, NJ, USA: Prentice Hall, 2001, pp. 273–333.

[26] S. Young, *The HTK Book*. Cambridge, U.K.: Cambridge Univ. Press, 2006, pp. 156–157.

[27] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*. Boston, MA, USA: Springer, 2009, pp. 659–663.

[28] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-97-021, 1998.

[29] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.

[30] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2006, pp. 97–100.

[31] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[32] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 609–616.

[33] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

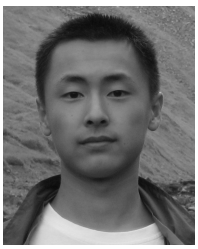
[34] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[35] D. Donoho, V. Stodden, and Y. Tsaig, *About SparseLab(v2.1)*. Stanford, CA, USA: Stanford Univ., 2007.

[36] J. Ortega-Garcia, J. Gonzalez-Rodriguez, and V. Marrero, "AHUMADA: A large speech corpus in Spanish for speaker characterization and identification," *Speech Commun.*, vol. 31, no. 2, pp. 255–264, Jun. 2000.

[37] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[38] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik, "What size test set gives good error rate estimates?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 52–64, Jan. 1998.



Yuechi Jiang (S'18) received the B.Eng. degree in electronic engineering from the Chinese University of Hong Kong in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include acoustic signal processing and pattern recognition.

Frank H. F. Leung (M'92–SM'03) received the B.Eng. and Ph.D. degrees in electronic engineering from The Hong Kong Polytechnic University, in 1988 and 1992, respectively. He is currently an Associate Professor with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. He is an active researcher who has published over 210 research papers on computational intelligence, machine learning, control, and power electronics. He is currently involved in the Research and Development on Intelligent Signal Processing, Systems, and Robotics. He has been serving as an editor, a guest editor, and a reviewer for international journals and helping the organization of many international conferences. He is currently an Executive Committee Member of IEEE Hong Kong Chapter of Signal Processing. He is also a Chartered Engineer and a Corporate Member of the Institution of Engineering and Technology, U.K., and the Hong Kong Institution of Engineers.