The following publication Y. Jiang and F. H. F. Leung, "Discriminative Collaborative Representation and Its Application to Audio Signal Classification," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 31-36 is available at https://doi.org/10.1109/ICPR.2018.8546024.

Discriminative Collaborative Representation and Its Application to Audio Signal Classification

Yuechi Jiang
Electronic and Information Engineering
The Hong Kong Polytechnic University
Hong Kong, China
yuechi.jiang@connect.polyu.hk

Frank H. F. Leung
Electronic and Information Engineering
The Hong Kong Polytechnic University
Hong Kong, China
frank-h-f.leung@polyu.edu.hk

***Abstract*—In this paper, we propose Discriminative Collaborative Representation (DCR) as an extension to Collaborative Representation (CR), by adding an extra discriminative term to the original formulation of CR. In the literature, both CR and Sparse Representation (SR) have been shown to be good in signal classification. Compared to SR, CR is more computationally efficient, but does not give obvious performance improvement. Therefore, we propose DCR, which aims at improving the performance of CR in signal classification. Besides, we extend DCR to Kernel DCR (KDCR), which generalizes DCR by introducing kernel functions. Comparisons among SR, CR and DCR are made in doing two audio signal classification tasks. Experimental results show that DCR can outperform CR and SR in both classification tasks, which demonstrates the effectiveness of our proposed DCR and the usefulness of the extra discriminative term.**

***Keywords*—Sparse representation; collaborative representation; discriminative collaborative representation; audio signal classification**

I. Introduction

Sparse Representation (SR) has been widely used for image and audio signal classification [1]. Regarding image classification, it has been applied to handwritten digit classification [2], vehicle classification [3], and face recognition [4]-[7]. Regarding audio classification, it has been applied to music genre classification [8][9], speech recognition [10][11], speaker recognition [12][13], and audio recording device identification [14]-[18].

In the perspective of SR, a signal is approximated by a linear combination of the basis vectors in a dictionary, and the SR of a signal is namely the collection of the coefficients in the linear combination. In the formulation of SR, an L1-norm based regularization term is usually incorporated. However, incorporating L1-norm regularization results in the inexistence of an analytic solution for SR. In order to calculate SR, people have to resort to some optimization techniques, such as Matching Pursuit (MP) and Basis Pursuit (BP) [2], or Homotopy and Augmented Lagrange Multiplier (ALM) [19].

Due to the difficulty of L1-norm regularization, in [20], the Collaborative Representation (CR) is proposed as an alternative to SR, where L2-norm regularization is incorporated. Owing to the incorporation of L2-norm instead of L1-norm, an analytic solution exists for CR, which makes CR computationally more efficient than SR. Yet, the performances of CR and SR are similar [20].

In this paper, we propose Discriminative Collaborative Representation (DCR), which introduces an extra discriminative term to the original formulation of CR, with an aim to improve the performance. This extra discriminative term controls the level of collaboration among different basis vectors in the dictionary, and therefore controls the discriminative ability of DCR. We then derive Kernel DCR (KDCR), which generalizes DCR by introducing different kernels. The performances of DCR, CR and SR are compared in two audio signal classification tasks, and experimental results show that the proposed DCR can outperform CR and SR. To process DCR, CR and SR, we propose Residual-based Classifier (RC) and then extend it to Kernel RC (KRC), which is a generalization of Sparse Representation-based Classifier (SRC) in [4] and Collaborative Representation-based Classifier (CRC) in [20]. Moreover, we also compare RC with Support Vector Machine (SVM), which is a widely used classifier in audio signal classification [12][13][21]-[27].

In this paper we consider two microphone identification tasks, which aim at identifying the recording microphone based on the recorded speech signal. Gaussian Supervector (GSV) is used as the feature vector, which is then used to calculate SR, CR and DCR. GSV has been widely used in different audio signal classification tasks, such as speaker recognition [12][21][26][27] and recording device identification [16][18][22]-[24]. An overview of the audio signal classification procedure is depicted in Fig. 1.

This paper is organized as follows. In Section II, the formulation of SR and CR is given. In Section III, the formulation of the proposed DCR and KDCR is given, together

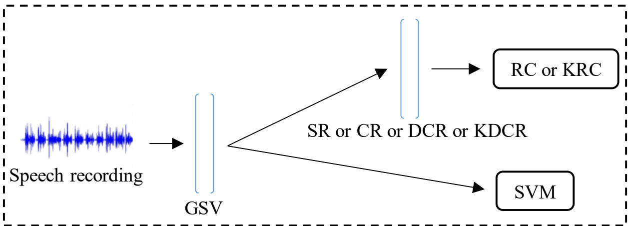

Fig. 1. Overview of audio signal classification procedure.

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project Account Code: RUG7).

with the corresponding RC and KRC. In Section IV, the audio datasets are briefly described. In Section V, the audio signal classification results are presented and discussed. In Section VI, a conclusion is drawn.

II. SR AND CR

Suppose we have a dictionary B as given by (1), which is a matrix comprising N basis vectors, whose i-th column vector b_i is namely the i-th basis vector. For a given feature vector x_n, its corresponding SR y_n is obtained by solving the optimization problem defined in (2), where $\|y\|_1$ is the L1-norm [4].

$$B = \begin{bmatrix} b_1 & b_2 & \cdots & b_N \end{bmatrix} \tag{1}$$

$$y_n = \arg\min_y \|y\|_1 \quad \text{subject to} \quad x_n = By \tag{2}$$

When the basis vectors b_i and the input feature vector x_n are noisy, instead of seeking the solution to (2), we can solve (3), which is the so-called unconstrained basis pursuit denoising (BPDN) problem, where λ is a weighting parameter [19].

$$y_n = \arg\min_y \frac{1}{2}\|x_n - By\|_2^2 + \lambda\|y\|_1 \tag{3}$$

Instead of incorporating the L1-norm, for a given feature vector x_n, its corresponding CR y_n is obtained by solving the optimization problem defined in (4). It can be shown that an analytical solution exists for CR, which is given by (5) [20].

$$y_n = \arg\min_y \|x_n - By\|_2^2 + \lambda\|y\|_2^2 \tag{4}$$

$$y_n = \left(B^T B + \lambda I\right)^{-1} B^T x_n \tag{5}$$

III. DCR

A. Discriminative Collaborative Representation

Suppose we have a dictionary B whose basis vectors are associated with different classes, then B can be denoted as a combination of C sub-dictionaries as given by (6), where C is the total number of sub-dictionaries (classes) and N_c is the number of basis vectors associated with class c. In this paper, the basis vectors are the training feature vectors, thus each basis vector (i.e. each training vector) is associated with a class.

$$\begin{aligned} &B = \begin{bmatrix} B^{(1)} & B^{(2)} & \cdots & B^{(C)} \end{bmatrix} \\ &\text{where} \quad B^{(c)} = \begin{bmatrix} b_1^{(c)} & b_2^{(c)} & \cdots & b_{N_c}^{(c)} \end{bmatrix} \end{aligned} \tag{6}$$

The i-th basis vector $b_i^{(c)}$ in $B^{(c)}$ is the $(N_1+\ldots+N_{c-1}+i)$-th basis vector $b_{N_1+\cdots+N_{c-1}+i}$ in B. Then we have (7), where $y_n^{(c)}$ is an $N_c\times 1$ vector comprising those elements in y_n that is only corresponding to the basis vectors in sub-dictionary $B^{(c)}$, and $\left(y_n^{(c)}\right)_i$ is the i-th element in $y_n^{(c)}$.

$$\begin{aligned} &N = \sum_{c=1}^{C} N_c \quad , \quad y_n^T = \begin{bmatrix} y_n^{(1)^T} & y_n^{(2)^T} & \cdots & y_n^{(C)^T} \end{bmatrix} \\ &B^{(c)} y_n^{(c)} = \sum_{i=1}^{N_c} b_i^{(c)} \left(y_n^{(c)}\right)_i \\ &By_n = \sum_{i=1}^{N} b_i \left(y_n\right)_i = \sum_{c=1}^{C} B^{(c)} y_n^{(c)} \end{aligned} \tag{7}$$

Different from CR which is obtained through solving the minimization problem defined in (4), the Discriminative Collaborative Representation (DCR) y_n is obtained through minimizing the objective function J_n defined in (8). The last term $J_n^{(3)}$ is a discriminative term. As can be seen from (8), $J_n^{(3)}$ represents the variance of the reconstructed signals based on different sub-dictionaries. The smaller the value of $J_n^{(3)}$, the smaller this variance will be, and the more collaboratively the basis vectors from different sub-dictionaries will work.

$$\begin{aligned} J_n &= \|x_n - By_n\|_2^2 + \lambda_1\|y_n\|_2^2 + \lambda_2 \sum_{i=1}^{C} \left\| B^{(i)} y_n^{(i)} - \frac{1}{C}\sum_{j=1}^{C} B^{(j)} y_n^{(j)} \right\|_2^2 \\ &= J_n^{(1)} + J_n^{(2)} + J_n^{(3)} \end{aligned} \tag{8}$$

where

$$\begin{aligned} &J_n^{(1)} = \|x_n - By_n\|_2^2 \quad , \quad J_n^{(2)} = \lambda_1\|y_n\|_2^2 \\ &J_n^{(3)} = \lambda_2 \sum_{i=1}^{C} \left\| B^{(i)} y_n^{(i)} - \frac{1}{C}\sum_{j=1}^{C} B^{(j)} y_n^{(j)} \right\|_2^2 \end{aligned} \tag{9}$$

Then the solution to (8) is obtained by setting the derivative of J_n to zero, as given by (10).

$$\frac{\partial J_n}{\partial y_n} = \frac{\partial J_n^{(1)}}{\partial y_n} + \frac{\partial J_n^{(2)}}{\partial y_n} + \frac{\partial J_n^{(3)}}{\partial y_n} = 0 \tag{10}$$

The derivative of J_n is obtained by summing up the derivatives of $J_n^{(1)}$, $J_n^{(2)}$ and $J_n^{(3)}$. The derivatives of $J_n^{(1)}$ and $J_n^{(2)}$ can be easily calculated as given by (11) and (12).

$$\frac{\partial J_n^{(1)}}{\partial y_n} = \frac{\partial \|x_n - By_n\|_2^2}{\partial y_n} = -2B^T\left(x_n - By_n\right) \tag{11}$$

$$\frac{\partial J_n^{(2)}}{\partial y_n} = \frac{\partial \lambda_1\|y_n\|_2^2}{\partial y_n} = 2\lambda_1 y_n \tag{12}$$

$J_n^{(3)}$ can be expanded as (13),

$$\begin{aligned} J_n^{(3)} &= \lambda_2 \sum_{i=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(i)} y_n^{(i)} - \lambda_2 \sum_{i=1}^{C} \frac{2}{C} y_n^{(i)^T} B^{(i)^T} \sum_{j=1}^{C} B^{(j)} y_n^{(j)} \\ &\quad + \lambda_2 \sum_{i=1}^{C} \frac{1}{C^2} \sum_{j=1}^{C}\sum_{k=1}^{C} y_n^{(j)^T} B^{(j)^T} B^{(k)} y_n^{(k)} \\ &= \lambda_2 \sum_{i=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(i)} y_n^{(i)} - \frac{\lambda_2}{C} \sum_{i=1}^{C}\sum_{j=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(j)} y_n^{(j)} \end{aligned} \tag{13}$$

Then in order to calculate the derivative of $J_n^{(3)}$ with respect to y_n, we first calculate the partial derivative of $J_n^{(3)}$ with respect to $y_n^{(c)}$ as given by (14).

$$\frac{\partial J_n^{(3)}}{\partial y_n^{(c)}}=\frac{\partial \lambda_2 \sum_{i=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(i)} y_n^{(i)}}{\partial y_n^{(c)}}-\frac{\partial \frac{\lambda_2}{C}\sum_{i=1}^{C}\sum_{j=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(j)} y_n^{(j)}}{\partial y_n^{(c)}}$$

$$=\frac{\partial \lambda_2 \sum_{i=1}^{C} y_n^{(i)^T} B^{(i)^T} B^{(i)} y_n^{(i)}}{\partial y_n^{(c)}}-\frac{\partial \frac{\lambda_2}{C}\sum_{j=1,j\neq c}^{C} y_n^{(c)^T} B^{(c)^T} B^{(j)} y_n^{(j)}}{\partial y_n^{(c)}}$$

$$-\frac{\partial \frac{\lambda_2}{C}\sum_{i=1,i\neq c}^{C} y_n^{(i)^T} B^{(i)^T} B^{(c)} y_n^{(c)}}{\partial y_n^{(c)}}-\frac{\partial \frac{\lambda_2}{C} y_n^{(c)^T} B^{(c)^T} B^{(c)} y_n^{(c)}}{\partial y_n^{(c)}} \tag{14}$$

$$=2\lambda_2 B^{(c)^T} B^{(c)} y_n^{(c)}-\frac{2\lambda_2}{C}\sum_{j=1,j\neq c}^{C} B^{(c)^T} B^{(j)} y_n^{(j)}-\frac{2\lambda_2}{C} B^{(c)^T} B^{(c)} y_n^{(c)}$$

$$=2\lambda_2 B^{(c)^T} B^{(c)} y_n^{(c)}-\frac{2\lambda_2}{C}\sum_{j=1}^{C} B^{(c)^T} B^{(j)} y_n^{(j)}$$

$$=2\lambda_2 B^{(c)^T} B^{(c)} y_n^{(c)}-\frac{2\lambda_2}{C} B^{(c)^T} B y_n$$

Then the derivative of $J_n^{(3)}$ with respect to y_n can be expressed in terms of the partial derivatives, as given by (15), where matrix L is defined in (16).

$$\frac{\partial J_n^{(3)}}{\partial y_n}=\begin{bmatrix}\frac{\partial J_n^{(3)}}{\partial y_n^{(1)}}\\ \frac{\partial J_n^{(3)}}{\partial y_n^{(2)}}\\ \vdots\\ \frac{\partial J_n^{(3)}}{\partial y_n^{(C)}}\end{bmatrix}=2\lambda_2\begin{bmatrix}B^{(1)^T}B^{(1)}y_n^{(1)}\\ B^{(2)^T}B^{(2)}y_n^{(2)}\\ \vdots\\ B^{(C)^T}B^{(C)}y_n^{(C)}\end{bmatrix}-\frac{2\lambda_2}{C}\begin{bmatrix}B^{(1)^T}\\ B^{(2)^T}\\ \vdots\\ B^{(C)^T}\end{bmatrix}By_n \tag{15}$$

$$=2\lambda_2 L y_n-\frac{2\lambda_2}{C}B^T B y_n$$

where

$$L=\begin{bmatrix}L^{(1)} & 0 & \cdots & 0\\ 0 & L^{(2)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & L^{(C)}\end{bmatrix}=\begin{bmatrix}B^{(1)^T}B^{(1)} & 0 & \cdots & 0\\ 0 & B^{(2)^T}B^{(2)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & B^{(C)^T}B^{(C)}\end{bmatrix} \tag{16}$$

After obtaining the derivatives of $J_n^{(1)}$, $J_n^{(2)}$ and $J_n^{(3)}$, (10) can then be expressed as (17). The solution to (17) is given by (18), which is then the formulation of DCR.

$$\left(-2B^T\left(x_n-By_n\right)\right)+2\lambda_1 y_n+\left(2\lambda_2 L y_n-\frac{2\lambda_2}{C}B^T B y_n\right)=0 \tag{17}$$

$$y_n=\left(\left(1-\frac{\lambda_2}{C}\right)B^T B+\lambda_1 I+\lambda_2 L\right)^{-1}B^T x_n \tag{18}$$

B. Kernel Discriminative Collaborative Representation

In (18), B^TB is an $N\times N$ matrix whose ij-th element is given by (19); the c-th block $L^{(c)}$ in L is an $N_c\times N_c$ matrix whose ij-th element is given by (20); B^Tx_n is an $N\times 1$ vector whose i-th element is given by (21).

$$\left(B^T B\right)_{ij}=b_i^T b_j \tag{19}$$

$$\left(L^{(c)}\right)_{ij}=\left(B^{(c)^T}B^{(c)}\right)_{ij}=b_i^{(c)^T}b_j^{(c)} \tag{20}$$

$$\left(B^T x_n\right)_i=b_i^T x_n \tag{21}$$

Since (19) ~ (21) only involve the inner product of two vectors, we can utilize a kernel function $k(a,b)=a^Tb$ to represent the inner product of two vectors a and b. On the other hand, the kernel function can also involve an implicit mapping of the original vector. For example, we can define $k(a,b)=\varphi(a)^T\varphi(b)$, and then φ will be the mapping function that maps the vector a from input feature space to another dimensional space. By utilizing a valid kernel function $k(.,.)$, (19) ~ (21) can be reformulated in terms of the kernel function as given by (22) ~ (24), where $\varphi(B)$ and $\varphi(B^{(c)})$ are merely two simplified notations defined in (25) and (26) respectively.

$$\left(\varphi(B)^T\varphi(B)\right)_{ij}=\varphi(b_i)^T\varphi(b_j)=k(b_i,b_j) \tag{22}$$

$$\left(\varphi\left(B^{(c)}\right)^T\varphi\left(B^{(c)}\right)\right)_{ij}=\varphi\left(b_i^{(c)}\right)^T\varphi\left(b_j^{(c)}\right)=k\left(b_i^{(c)},b_j^{(c)}\right) \tag{23}$$

$$\left(\varphi(B)^T\varphi(x_n)\right)_i=\varphi(b_i)^T\varphi(x_n)=k(b_i,x_n) \tag{24}$$

where

$$\varphi(B)=\left[\varphi(b_1)\quad\varphi(b_2)\quad\cdots\quad\varphi(b_N)\right] \tag{25}$$

$$\varphi\left(B^{(c)}\right)=\left[\varphi\left(b_1^{(c)}\right)\quad\varphi\left(b_2^{(c)}\right)\quad\cdots\quad\varphi\left(b_{N_c}^{(c)}\right)\right] \tag{26}$$

To formulate Kernel Discriminative Collaborative Representation (KDCR), we define an $N\times N$ matrix K_{BB} as given by (27), an $N\times 1$ vector K_{Bx_n} as given by (28), and an $N\times N$ matrix K_{CC} as given by (29) and (30).

$$K_{BB}=\begin{bmatrix}k(b_1,b_1) & k(b_1,b_2) & \cdots & k(b_1,b_N)\\ k(b_2,b_1) & k(b_2,b_2) & \cdots & k(b_2,b_N)\\ \vdots & \vdots & \ddots & \vdots\\ k(b_N,b_1) & k(b_N,b_2) & \cdots & k(b_N,b_N)\end{bmatrix} \tag{27}$$

$$K_{Bx_n}=\begin{bmatrix}k(b_1,x_n)\\ k(b_2,x_n)\\ \vdots\\ k(b_N,x_n)\end{bmatrix} \tag{28}$$

$$K_{CC}=\begin{bmatrix}K_{CC}^{(1)} & 0 & \cdots & 0\\ 0 & K_{CC}^{(2)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & K_{CC}^{(C)}\end{bmatrix} \tag{29}$$

where

$$K_{CC}^{(c)} = \begin{bmatrix} k(b_1^{(c)}, b_1^{(c)}) & k(b_1^{(c)}, b_2^{(c)}) & \cdots & k(b_1^{(c)}, b_{N_c}^{(c)}) \\ k(b_2^{(c)}, b_1^{(c)}) & k(b_2^{(c)}, b_2^{(c)}) & \cdots & k(b_2^{(c)}, b_{N_c}^{(c)}) \\ \vdots & \vdots & \ddots & \vdots \\ k(b_{N_c}^{(c)}, b_1^{(c)}) & k(b_{N_c}^{(c)}, b_2^{(c)}) & \cdots & k(b_{N_c}^{(c)}, b_{N_c}^{(c)}) \end{bmatrix} \quad (30)$$

Having defined K_{BB}, K_{Bxn} and K_{CC}, (18) can then be reformulated as (31), which is the formulation of KDCR. As a matter of fact, DCR is just KDCR with the linear kernel.

$$y_n = \left(\left(1 - \frac{\lambda_2}{C} \right) K_{BB} + \lambda_1 I + \lambda_2 K_{CC} \right)^{-1} K_{Bx_n} \quad (31)$$

C. Residual-based Classifier and Its Kernel Version

Both Sparse Representation-based Classifier (SRC) and Collaborative Representation-based Classifier (CRC) are residual-based classifiers, which means for each class, SRC or CRC calculates a residual, and the input feature vector x_n will be classified to the class that has the minimum residual. On using DCR, the residual $r_n^{(c)}$ for the c-th class is given by (32).

$$\begin{aligned} r_n^{(c)} &= \left\| x_n - B^{(c)} y_n^{(c)} \right\|_2^2 = \left(x_n - B^{(c)} y_n^{(c)} \right)^T \left(x_n - B^{(c)} y_n^{(c)} \right) \\ &= x_n^T x_n - 2 x_n^T B^{(c)} y_n^{(c)} + y_n^{(c)T} B^{(c)T} B^{(c)} y_n^{(c)} \end{aligned} \quad (32)$$

On using KDCR, the residual $r_{n,\varphi}^{(c)}$ is given by (33), where K_{Cxn} is an $N_c \times 1$ vector, whose i-th element is given by (34).

$$\begin{aligned} r_{n,\varphi}^{(c)} &= \left\| \varphi(x_n) - \varphi\left(B^{(c)}\right) y_n^{(c)} \right\|_2^2 \\ &= \left(\varphi(x_n) - \varphi\left(B^{(c)}\right) y_n^{(c)} \right)^T \left(\varphi(x_n) - \varphi\left(B^{(c)}\right) y_n^{(c)} \right) \\ &= \varphi(x_n)^T \varphi(x_n) - 2 \varphi(x_n)^T \varphi\left(B^{(c)}\right) y_n^{(c)} \\ &\quad + y_n^{(c)T} \varphi\left(B^{(c)}\right)^T \varphi\left(B^{(c)}\right) y_n^{(c)} \\ &= k(x_n, x_n) - 2 \left(\varphi\left(B^{(c)}\right)^T \varphi(x_n) \right)^T y_n^{(c)} + y_n^{(c)T} K_{CC}^{(c)} y_n^{(c)} \\ &= k(x_n, x_n) - 2 K_{Cx_n}^T y_n^{(c)} + y_n^{(c)T} K_{CC}^{(c)} y_n^{(c)} \end{aligned} \quad (33)$$

where

$$\left(K_{Cx_n} \right)_i = \varphi\left(b_i^{(c)}\right)^T \varphi(x_n) = k\left(b_i^{(c)}, x_n\right) \quad (34)$$

For convenience, the classification procedure of Residual-based Classifier (RC) is described in Algorithm 1, which summarizes SRC and CRC; the classification procedure of Kernel Residual-based Classifier (KRC) is described in Algorithm 2, which generalizes RC by introducing the kernel.

IV. Experimental Settings and Datasets

In our experiments, Gaussian Supervector (GSV) is used as the input feature vector for the calculation of SR, CR, DCR and KDCR, which is obtained by adapting a 32-component Universal Background Model (UBM) with a relevance factor of 20. More details of GSV can be found in [21]-[23]. On using KDCR, we adopt Gaussian kernel defined as $k(a,b) = e^{-(a-b)^T (a-b)/h}$, where h is the kernel parameter.

Algorithm 1: RC

1: Given the SR or CR or DCR y_n corresponding to the input feature vector x_n, calculate the residual $r_n^{(c)}$ with respect to each class c as follows,
$r_n^{(c)} = \left\| x_n - B^{(c)} y_n^{(c)} \right\|_2^2$

2: The input feature vector x_n is then classified to the class that has the minimum residual, i.e.
$class(x_n) = \arg\min_c r_n^{(c)}$ where $c \in \{1, 2 \ldots C\}$

Algorithm 2: KRC

1: Given the KDCR y_n corresponding to the input feature vector x_n, calculate the residual $r_{n,\varphi}^{(c)}$ with respect to each class c as follows,
$r_{n,\varphi}^{(c)} = k(x_n, x_n) - 2 K_{Cx_n}^T y_n^{(c)} + y_n^{(c)T} K_{CC}^{(c)} y_n^{(c)}$

2: The input feature vector x_n is then classified to the class that has the minimum residual, i.e.
$class(x_n) = \arg\min_c r_{n,\varphi}^{(c)}$ where $c \in \{1, 2 \ldots C\}$

The audio datasets are Ahumada-25 and Gaudi-25, which are parts of AHUMADA [28]. We divide each dataset into a training set, a testing set, and a UBM set. The training set is used as the dictionary, while the testing set is used to test the performance of SR, CR, DCR and KDCR. The UBM set is used to calculate GSV. In Ahumada-25, we choose 4 different microphone models as listed in Table I. Totally there are 960 speech recordings used for training and 1040 speech recordings used for testing. Another 599 microphone speech recordings are used for UBM. In Gaudi-25, we choose 5 different microphone models as listed in Table II. Totally there are 1200 speech recordings used for training and 1280 speech recordings used for testing. Another 744 microphone speech recordings are used for UBM.

V. Experimental Results and Discussions

In the first part of this section, we compare the performances of linear SVM, SR, CR and DCR in two microphone classification tasks, where SVM is implemented by LIBSVM [29]. In the second part of this section, we compare the performances of DCR and KDCR, where Gaussian kernel is employed on using KDCR. SR is obtained

TABLE I. AHUMADA-25 MICROPHONE SPEECH DATASET

Notation	Microphone Model	Number of Speech Recordings	
		Training	***Testing***
M1	AKG C410B Head Mounted	240	260
M2	AKH D80S Desktop	240	260
M3	SONY ECM 66B Lapel	240	260
M4	TARGET Lapel	240	260
UBM	All the models	599	

TABLE II. GAUDI-25 MICROPHONE SPEECH DATASET

Notation	Microphone Model	Number of Speech Recordings	
		Training	***Testing***
M1	AKG C410 Desktop	240	260
M2	AKG Tripower Desktop	240	260
M3	AKH D80S Desktop	240	260
M4	SONY ECM 66B Lapel	240	260
M5	TARGET CPT3GX Desktop	240	240
UBM	All the models	744	

using Basis Pursuit (BP) algorithm [30] implemented by SparseLab [31]. The SR calculated using (2) is called SR (BP), while the SR calculated using (3) is called SR (BPDN). On using SR, CR and DCR, different parameters are evaluated.

The classification results of SR, CR, DCR and SVM are plotted in Fig. 2. It can be seen that, SR (BP) performs worse than SR (BPDN), as the input feature vector (i.e. GSV) is usually noisy and therefore the regularization term in (3) is necessary for robustness. In addition, SR (BPDN) can perform better than SVM, and CR can work a little better than SR (BPDN). Moreover, the proposed DCR can outperform SR, CR and SVM, when the discriminative parameter λ_2 is carefully chosen. This demonstrates the necessity and usefulness of adding the discriminative term to the original formulation of CR, which makes the representation more robust and flexible. As can be seen from (8), the regularization term $J_n^{(2)}$ controls the level of sparseness of DCR, while the discriminative term $J_n^{(3)}$ controls the level of collaboration of the coefficients in DCR. The larger the value of λ_1, the sparser the DCR will be; the larger the value of λ_2, the more similar the reconstructed signals based on different sub-dictionaries will be. In a word, the regularization term controls the relationship of the coefficients in DCR, while the discriminative term controls the relationship of the coefficients in different sub-dictionaries. The two terms interact with each other.

The classification results of using DCR and KDCR are illustrated in Figs. 3 and 4. The major observation is that, Gaussian kernel does not help, as compared to linear kernel. This means an extra feature mapping may be useless. On increasing the value of the regularization parameter λ_1, the performances of KDCR with different values of the discriminative parameter λ_2 tend to converge. The larger the kernel parameter h, the faster the performances will converge. This convergence phenomenon also demonstrates the usefulness of the regularization term λ_1 in improving the robustness of the KDCR.

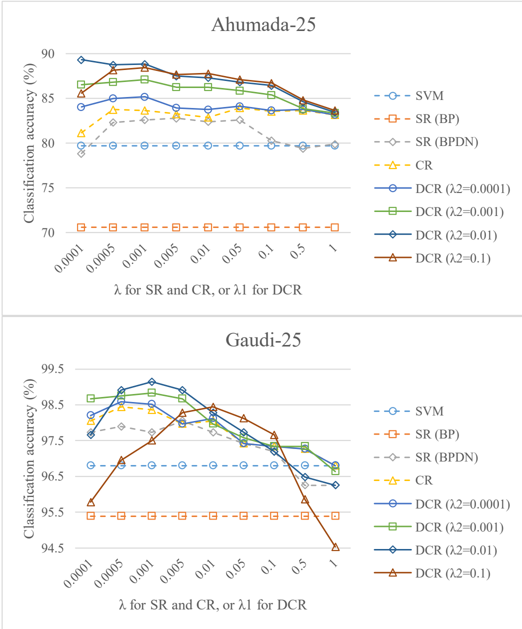

Fig. 2. Comparison among SVM, SR, CR and DCR on Ahumada-25 and Gaudi-25.

VI. CONCLUSION

In this paper, we propose Discriminative Collaborative Representation (DCR), which is the extension and improvement of Collaborative Representation (CR). In the literature, Sparse Representation (SR) and CR have shown to be good representations for signal classification, especially for image classification and audio classification, while CR is more computationally efficient than SR. However, compared to SR, CR merely improves the computational efficiency but not classification performance. Facing this situation, we propose DCR, which introduces an extra discriminative term to the original formulation of CR. This discriminative term controls the discriminative ability of DCR, making it more robust and flexible than CR. Experimental results on two audio signal classification tasks demonstrate that the proposed DCR can outperform CR as well as SR. In addition, we also develop the kernel version of DCR (KDCR), which is the generalization of DCR (as DCR is KDCR with the linear kernel). However, KDCR with Gaussian kernel does not offer improvement over DCR, which means that an extra feature mapping before classification seems unnecessary for DCR. Nevertheless, KDCR provides a way to include an implicit feature mapping before classification, which may be useful in some situation.

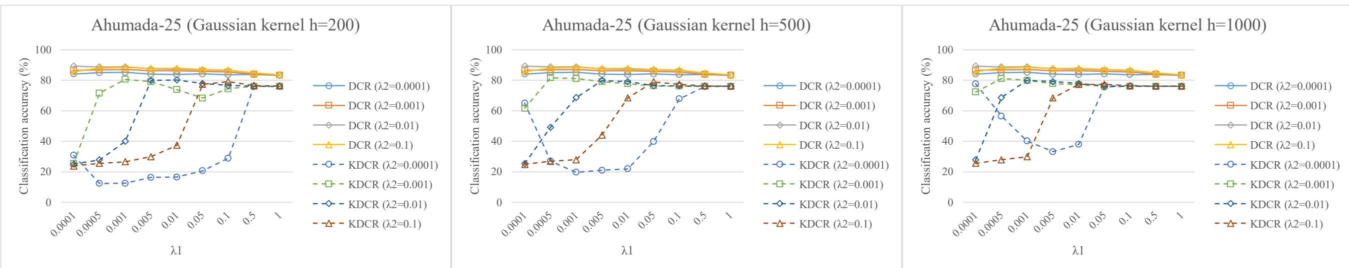

Fig. 3. Comparison between DCR and KDCR on Ahumada-25.

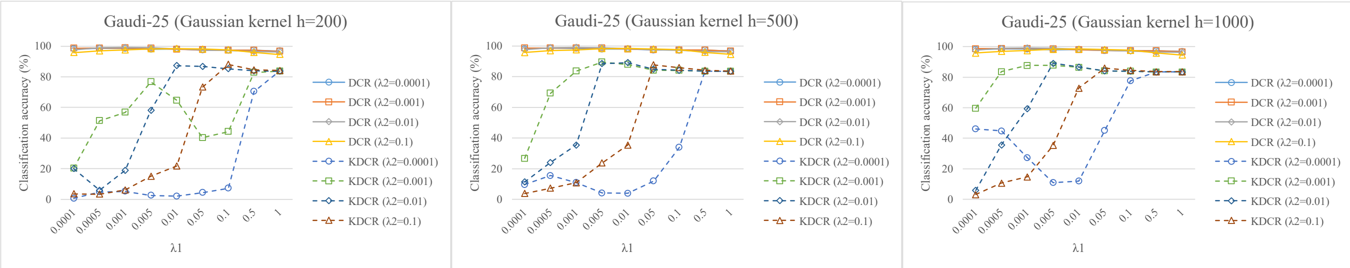

Fig. 4. Comparison between DCR and KDCR on Gaudi-25.

REFERENCES

[1] J. Wright *et al.*, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031-1044, 2010.

[2] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 609-616.

[3] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259-2272, 2011.

[4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, 2009.

[5] A. Wagner *et al.*, "Toward a practical face recognition system: robust alignment and illumination by sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 372-386, 2012.

[6] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 625-632.

[7] C. Y. Lu, H. Min, J. Gui, L. Zhu, and Y. K. Lei, "Face recognition via weighted sparse representation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 111-116, 2013.

[8] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music genre classification via sparse representations of auditory temporal modulations," in *Proc. IEEE European Signal Processing Conference (EUSIPCO)*, 2009, pp. 1-5.

[9] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce, "Music genre classification via joint sparse low-rank representation of audio features," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1905-1917, 2014.

[10] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067-2080, 2011.

[11] T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-based sparse representation features: from TIMIT to LVCSR," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2598-2613, 2011.

[12] I. Naseem, R. Togneri, and M. Bennamoun, "Sparse representation for speaker identification," in *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, 2010, pp. 4460-4463.

[13] M. Li, X. Zhang, Y. Yan, and S. Narayanan, "Speaker verification using sparse representations on total variability i-vectors," in *Proc. 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 2729-2732.

[14] Y. Panagakis and C. Kotropoulos, "Automatic telephone handset identification by sparse representation of random spectral features," in *Proc. ACM Multimedia and Security*, 2012, pp. 91-96.

[15] C. Kotropoulos, "Telephone handset identification using sparse representations of spectral feature sketches," in *Proc. IEEE Int. Workshop on Biometrics and Forensics (IWBF)*, 2013, pp. 1-4.

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

[17] 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. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 2079-2083.

[18] L. Zou, Q. He, and J. Wu, "Source cell phone verification from speech recordings using sparse representation," *Digital Signal Processing*, vol. 62, pp. 125-136, 2017.

[19] A. Y. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma, "Fast l1-minimization algorithms for robust face recognition," *IEEE Trans. on Image Processing*, vol. 22, no. 8, pp. 3234-3246, 2013.

[20] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: which helps face recognition," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 471-478.

[21] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308-311, 2006.

[22] D. G. Romero and C. Y. E. Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010, pp. 1806-1809.

[23] Y. Jiang and F. H. F. Leung, "Using regularized Fisher discriminant analysis to improve the performance of Gaussian supervector in session and device identification," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2017, pp. 705-712.

[24] Y. Jiang and F. H. F. Leung, "Using double regularization to improve the effectiveness and robustness of Fisher discriminant analysis as a projection technique," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2018.

[25] Y. Jiang and F. H. F. Leung, "Mobile phone identification from speech recordings using weighted support vector machine," in *Proc. IEEE 42nd Annual Conf. on Industrial Electronics (IECON)*, 2016, pp. 963-968.

[26] Y. Jiang and F. H. F. Leung, "Generalized Fisher discriminant analysis as a dimensionality reduction technique," in *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR)*, 2018.

[27] Y. Jiang and F. H. F. Leung, "The scalable version of probabilistic linear discriminant analysis and its potential as a classifier for audio signal classification," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, 2018.

[28] J. O. Garcia, J. G. Rodriguez, and V. M. Aguiar, "AHUMADA: a large speech corpus in Spanish for speaker characterization and identification," *Speech Communication*, vol. 31, no. 2, pp. 255-264, 2000.

[29] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011.

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

[31] D. Donoho, V. Stodden, and Y. Tsaig, *About SparseLab(v2.1)*, Stanford, 2007.