Multimed Tools Appl (2018) 77:713–739
DOI 10.1007/s11042-016-4287-0

# Semi-supervised minimum redundancy maximum relevance feature selection for audio classification

**Xu -Kui Yang**[1] **· Liang He**[2] **· Dan Qu**[1] **· Wei-Qiang Zhang**[2]

Received: 26 June 2016 / Revised: 26 October 2016 / Accepted: 20 December 2016 /
Published online: 30 December 2016


**Abstract** It is still a changing problem of choosing the most relevant ones from multiple features for their specific machine learning tasks. However, feature selection provides an effective solution to it, which aims to choose the most relevant and least redundant features for data analysis. In this paper, we present a feature selection algorithm termed as semi-supervised minimum redundancy maximum relevance. The relevance is measured by a semi-supervised filter score named constraint compensated Laplacian score, which takes advantage of the local geometrical structures of unlabeled data and constraint information from labeled data. The redundancy is measured by a semi-supervised Gaussian mixture model-based Bhattacharyya distance. The optimal feature subset is selected by maximizing feature relevance and minimizing feature redundancy simultaneously. We apply our algorithm in audio classification task and compare it with other known feature selection methods. Experimental results further prove that our algorithm can lead to promising improvements.

**Keywords** Audio classification · Semi-supervised feature selection · Minimal redundancy · Maximal relevance · Locality preserving · Constraint information · Bhattacharyya distance

## 1 Introduction

In the field of machine learning, especially in audio related fields, there are numerous features for choices in model construction for each specific application. However, it is still a challenging problem that how we can choose the most relevant features to construct a more effective

✉ Xu -Kui Yang
gzyangxk@gmail.com

Dan Qu
qudanqudan@sina.com

[1] Zhengzhou Information Science and Technology Institute, Zhengzhou, China

[2] Department of Electronic Engineering, Tsinghua University, Beijing, China

model. It may need very professional background knowledge, yet it can be solved by automatic feature selection [39]. Feature selection produces features which are more discriminative or easier for statistical modeling and hence promises higher accuracy by removing the irrelevant and redundant features [17].

From class information utilizations, feature selection can be divided into three categories: supervised [30], unsupervised [9, 10], or semi-supervised [33]. Supervised approaches need lots of labeled data, and they are apt to ignore the internal structures of data whereas focus too much on label information. Due to the absence of class labels, unsupervised feature selection fails to extract more discriminative features which may yield worse performance. Semi-supervised feature selection focuses on solving the small-labeled-sample problem [37], where the amount of unlabeled data is much larger than that of labeled data. This type of algorithm has attracted more attentions for its comprehensive considerations of label information and data intrinsic structure characteristics.

From the perspective of selection strategy, feature selection are categorized as filter, wrapper and embedded. The filter methods use scores or confidences to evaluate the relevance of features to the learning tasks. There are various kind of filter based algorithms, for example, Laplacian score (LS) [18], constraint score (CS) [36] and constraint compensated Laplacian score (CCLS) [34]. The wrapper approaches evaluate the different subsets of features by some specific learning algorithms and select the one with the best performance. The embedded model techniques search the most relevant and effective features for models while their constructions. The most common embedded methods are regularization-based [11], e.g., C4.5 [24] and LARS [12].

Since the filter methods are irrelevant to any specific classification or learning methods, they have been widely used for their better generalization properties. However, it may be very simple to select the top-ranked features only based on the feature relevance, because these features could be correlated among themselves. In other words, the set of selected features contains a certain redundancy, and this redundancy will degrade the learning performances and complicate the models. Several studies have addressed influences of such redundancy [2, 8, 35]. Among them, the most famous one is minimum redundancy maximum relevancy (mRMR) algorithm [23] in which the features are selected by simultaneously optimizing the minimum redundancy and the maximum relevance conditions. For mRMR algorithm, either redundancy between features or relevance between the features and the corresponding classes is measured by mutual information (MI). However, when the values of feature vectors are continuous, both types of MI are difficult to compute because it needs to calculate integral which limits the application ranges of mRMR algorithm most in discrete data like genes.

In this paper, we propose a new feature selection algorithm which selects the optimal feature set similar to mRMR. Rather than using MI to measure relevance and redundancy, a novel semi-supervised relevance measurement named constraint compensated Laplacian score (CCLS) is proposed and a semi-supervised Gaussian mixture model (GMM)-based Bhattacharyya distance [5] is used as the score of minimum redundancy. In traditional Laplacian score, the features are evaluated according to their locality preserving abilities. Compared to unsupervised constructions of local and global structures in Laplacian score, CCLS uses constraint information generated from a small amount of labeled data to compensate these constructions. The GMM-based Bhattacharyya distance first classifies the unlabeled data in training dataset according to the labeled data, and then a GMM is used to model the data of each class. Finally, the redundancy is measured by the Bhattacharyya distance

calculated from these GMMs. Because the relevance and redundancy measurements in our algorithm are both semi-supervised, our algorithm is termed as semi-Supervised minimum redundancy maximum relevance (SSMRMR).

We use SSMRMR in audio classification. In this application, there are dozens of features to be utilized and we have to pick up effective ones or their combinations. The experimental results proved that CCLS outperformed classical LS and CS and the GMM-based Bhattacharyya distance was superior to the correlation-based or mutual information-based redundancy measurements. Moreover, the SSMRMR could remove irrelevant features and improve classification accuracy significantly.

The outline of this paper is as follows: definitions and notations are given in Section 2. Section 3 enumerates several main methods used in feature selections. Then we present our SSMRMR algorithm in Section 4. Section 5 depicts the backgrounds of audio classifications, experimental setup and analysis results. Finally, the conclusion is given in Section 6.

## 2 Definitions and notations

In this section, we will provide basic terminologies and notations which are necessary for the understanding of subsequent algorithms.

In this work, let the training dataset with $N$ instances be $\mathrm{X} = \{\mathbf{x}_i \in \mathbb{R}^M | \, i = 1, 2, \cdots, N\}$. Let $F_1, F_2, \cdots, F_M$ denote the $M$ features of X and $\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_M$ denote the corresponding feature vectors. Let $f_{ri}$ denote the $r$-th feature of the $i$-th instance $\mathbf{x}_i$, $i = 1, 2, \cdots, N$, $r = 1, 2, \cdots, M$. More specifically,

$$\mathrm{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{MN} \end{bmatrix} = \left[\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, \cdots, \mathbf{f}_M^{\mathrm{T}}\right]^{\mathrm{T}} \tag{1}$$

which means $\mathbf{f}_r = [f_{r1}, f_{r2}, \cdots, f_{rN}]^{\mathrm{T}}$ and $\mathbf{x}_i = [f_{1i}, f_{2i}, \cdots, f_{Mi}]^{\mathrm{T}}$.

In semi-supervised learning, the training dataset X can be divided into two subsets. The first contains data $\mathrm{X}^l = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L\}$ with labels $\mathrm{Y}^l = \{y_1, y_2, \cdots, y_L | \, y_i = 1, 2, \cdots, C\}$, where $C$ is the number of classes and $L$ is the number of labeled data. And the second one only has the unlabeled data $\mathrm{X}^u = \{\mathbf{x}_{L+1}, \mathbf{x}_{L+2}, \cdots, \mathbf{x}_N\}$.

Define $\mu_r^l = \sum_{i|\mathbf{x}_i \in \mathrm{X}^l} f_{ri}/L$ is the mean of the $r$-th feature of the labeled data. Define $\mu_r$ and $\mu_r^{(c)}$ be the $r$-th feature means of the whole dataset and the $c$-th class respectively, $\sigma_r^2$ and $\left(\sigma_r^{(c)}\right)^2$ denote its corresponding variances. $n_c$ is the number of instances corresponding to the class $c$.

For any pair of instances $(\mathbf{x}_i, \mathbf{x}_j)$ in $\mathrm{X}^l$, there are two types of constraints: must-link (ML) and cannot-link (CL). The ML constraint is constructed if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same class label, and the CL constraint is formed when $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different classes. Then, according to ML and CL constraints, the data are grouped into two sets $\Omega_{ML}$ and $\Omega_{CL}$, respectively.

From the consideration of data geometric structure, there are a set of pairwise instances similarity measures which can be used to represent the relationships between two instances. In

this paper, we choose the RBF kernel function to be the similarity measure for its unsupervised property. The similarity $w_{ij}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by:

$$w_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{2}$$

where, $\sigma$ is a constant and $\|\cdot\|^2$ is the square of Euclidian norm.

## 3 Related work

In this section, we shall list a collection of scores which are bases for the score functions of our framework. We illustrate both advantages and disadvantages of Laplacian score and constraint score. Moreover, the framework of mRMR is presented.

### 3.1 Laplacian score

Laplacian Score is a recently proposed unsupervised feature selection method [18]. The basic idea is to evaluate the features according to their locality preserving ability. If two data points are close to each other, they belong to the same class with high probability. So the local structure is more important than global structure in many machine learning problems, especially for classification tasks. The Laplacian score of the $r$-th feature is computed as follows:

$$L_r = \frac{\sum_{i,j}\left(f_{ri} - f_{rj}\right)^2 S_{ij}}{\sum_i (f_{ri} - u_r)^2 D_{ii}} \tag{3}$$

where, $u_r = \sum_{i=1}^{N} f_{ri}/N$ denotes the mean of the $r$-th feature of the whole data set, $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$, and $\mathbf{S}$ denotes the similarity matrix whose nonzero element is the RBF kernel function defined in Eq. (2):

$$S_{ij} = \begin{cases} w_{ij} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where, $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors which means that $\mathbf{x}_i$ is among $k$ nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the $k$ nearest neighbors of $\mathbf{x}_i$.

In the score function in Eq. (3), the numerator indicates the locality preserving power of the $r$-th feature, the smaller the better. The denominator is the estimated variance of the $r$-th feature, the bigger the better. Thus, the criterion of LS for choosing a good feature is to minimize the object function in Eq. (3).

Compared to other unsupervised feature selection algorithms [9, 10], the main advantage of LS is its powerful locality preserving ability which can be thought of as the degree a feature respects the nearest neighbor graph structure. However, there is some blindness when LS constructs the local structure of data space without supervised information.

### 3.2 Constraint score

Constraint Score is a supervised feature selection algorithm which need small amount of labeled data [36]. Firstly, the pairwise instances level constraints between any two data points,

ML and CL constraints, are generated using the data labels, and the score function $C_r$ is computed as follows using these pairwise constraints:

$$C_r = \frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\Omega_{ML}} \left(f_{ri} - f_{rj}\right)^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\Omega_{CL}} \left(f_{ri} - f_{rj}\right)^2} \tag{5}$$

In this score function, a good feature means that two ML-constraint point pair should be close to each other, CL-constraint point pair should be far away from each other. So the constraint score of the $r$-th feature should be minimized. CS feature selection algorithm is particularly applied to the cases where very few labeled training data are available. In these cases, CS can select a reliable feature subset only based on the limited training data. However, when there are large amount of unlabeled data in the training set, how to use these unlabeled samples to improve performance is still a challenge problem.

### 3.3 Minimum redundancy maximum relevancy

The mRMR algorithm focuses on MI-based feature selection. Given two random variables $z_1$ and $z_2$, suppose that $p(z_1)$, $p(z_2)$, and $p(z_1, z_2)$ are their marginal and joint probabilistic density functions. Their mutual information is defined as follows:

$$I(z_1, z_2) = \iint p(z_1, z_2) \log \frac{p(z_1, z_2)}{p(z_1)p(z_2)} dz_1 dz_2 \tag{6}$$

The mRMR feature set is obtained by minimum redundancy condition and maximum relevance condition simultaneously, either in quotient form:

$$\max_{\Lambda\subset\Omega} \left\{ \sum_{\mathbf{f}_i\in\Lambda} I(\mathrm{Y}, \mathbf{f}_i) \Big/ \left[ \frac{1}{|\Lambda|} \sum_{\mathbf{f}_i, \mathbf{f}_j\in\Lambda} I\left(\mathbf{f}_i, \mathbf{f}_j\right) \right] \right\} \tag{7}$$

or in difference form:

$$\max_{\Lambda\subset\Omega} \left\{ \sum_{\mathbf{f}_i\in\Lambda} I(\mathrm{Y}, \mathbf{f}_i) - \left[ \frac{1}{|\Lambda|} \sum_{\mathbf{f}_i, \mathbf{f}_j\in\Lambda} I\left(\mathbf{f}_i, \mathbf{f}_j\right) \right] \right\} \tag{8}$$

where, $\Lambda$ is the features subset under seeking and $\Omega$ is the set of entire candidate features. $|\Lambda|$ is the number of features in $\Lambda$. $I(\mathrm{Y}, \mathbf{f}_i)$ is the MI between the feature $\mathbf{f}_i$ and its corresponding classes Y. $I(\mathbf{f}_i, \mathbf{f}_j)$ is MI between feature $\mathbf{f}_i$ and $\mathbf{f}_j$.

For discrete (categorical) feature variables, the MI is easy and straightforward to be calculated, because the integral operation is reduces to summation, and moreover the probability can be approximated by counting the instances of discrete variables in the data based on ML criterion.

However, it is hard to compute the MI when the feature variables are continuous. Because only based on a limited number of instances, it is difficult to compute the integral in the continuous space. To solve this problem, one can either discretize the continuous data before computing MI [28], or use density estimation method to estimate MI approximately [19].

## 4 Semi-supervised minimum redundancy maximum relevance feature selection

In the following sections, we present the score of maximum relevance (constraint compensated Laplacian score) and the score of minimum redundancy (GMM-based Bhattacharyya distance) in our framework for feature selection. We also present the objective function of our algorithm and its approximate solution.

### 4.1 Feature relevance

#### *4.1.1 Score function*

In order to take advantages of both LS and CS as well as to overcome their shortcomings, we propose the constraint compensated Laplacian score algorithm [34]. The score function, which should be minimized, is defined as follows:

$$\eta_r = \frac{\sum_{i,j}\left(f_{ri}-f_{rj}\right)^2\left(S_{ij}+\overline{\mathcal{N}}_{ij}\right)}{\Sigma_r+\Sigma_r^b-\Sigma_r^w} \tag{9}$$

where,

$$\overline{\mathcal{N}}_{ij} = \begin{cases} 1-w_{ij} & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ & \text{and } \left(\mathbf{x}_i,\mathbf{x}_j\right)\in\Omega_{ML}. \\ -\gamma w_{ij} & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ & \text{and } \left(\mathbf{x}_i,\mathbf{x}_j\right)\in\Omega_{CL}. \\ \lambda & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are not neighbors} \\ & \text{and } \left(\mathbf{x}_i,\mathbf{x}_j\right)\in\Omega_{ML}. \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where, $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors which means that $\mathbf{x}_i$ is among $k$ nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the $k$ nearest neighbors of $\mathbf{x}_i$. $\gamma$ and $\lambda$ are the parameters set using the empirical values of 0.9 and 0.5 respectively [34], $S_{ij}$ is the same as Eq. (4) and is computed using both labeled and unlabeled data. $\Sigma_r$ is the variance of the whole dataset X, $\Sigma_r^w$ and $\Sigma_r^b$ are inner-class variance and inter-class variance of the labeled dataset $\mathrm{X}^l$, respectively.

$$\Sigma_r = \sum_i (f_{ri}-\mu_r)^2 D_{ii} \tag{11}$$

$$\Sigma_r^b = \sum_c n_c\left(\mu_r^{(c)}-\mu_r^l\right)^2 \tag{12}$$

$$\Sigma_r^w = \sum_c n_c\left(\sigma_r^{(c)}\right)^2 \tag{13}$$

And let $\boldsymbol{\Psi}=[\eta_r|r=1,2,\cdots,M]$ be the relevancy vector which represent the interior relevance of feature series in each dimension.

#### 4.1.2 Spectral graph analysis

In this section, we can also give an alternative explanation based on spectral graph theory [7] for the score function described above. The basic ideal of CCLS is that: a "good" feature must have strong locality preserving power, and a good global structure. Strong locality preserving power means that the two local structures constructed by only using this feature or using the complete feature set are consistent. A good global structure means that the instances of different classes are far from each other while instances of the same class are close to each other [18].

For locality preserving power, we first construct a similarity matrix to model the local geometric structure in a semi-supervised method. And then the locality preserving power of one feature can be regard as the degree it respects the similarity matrix. The detailed procedures are as follows:

Firstly, we construct three graphs $\mathbf{G}$, $\mathbf{G}^M$, and $\mathbf{G}^C$ all with $N$ nodes, which represent the information of neighbors, must-link constraints and cannot-link constraints respectively. In these graphs, the $i$-th node corresponds to the $i$-th instance $\mathbf{x}_i$. We put an edge between node $i$ and node $j$ in $\mathbf{G}$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, i.e. if $\mathbf{x}_i$ is one of the $k$ nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is one of the $k$ nearest neighbors of $\mathbf{x}_i$, $G_{ij} = 1$ . We put an edge between node $i$ and node $j$ in $\mathbf{G}^M$ if there is a must-link constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, which means if $(\mathbf{x}_i, \mathbf{x}_j) \in \Omega_{ML}$, $G^M_{ij} = 1$. Similarly, We put an edge between node $i$ and node $j$ in $\mathbf{G}^C$ if there is a cannot-link constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, namely if $(\mathbf{x}_i, \mathbf{x}_j) \in \Omega_{CL}$, $G^C_{ij} = 1$.

Once these graphs are constructed, we define the similarity matrix $\mathbf{S}$ whose elements are defined as follow:

$$\mathcal{S}_{ij} = \begin{cases} 1 & \text{if } G_{ij} = 1 \text{ and } G^M_{ij} = 1. \\ (1-\gamma)w_{ij} & \text{if } G_{ij} = 1 \text{ and } G^C_{ij} = 1. \\ \lambda & \text{if } G_{ij} = 0 \text{ and } G^M_{ij} = 1. \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

And define the Laplacian matrix as $\mathbf{L} = \mathbf{D} - S$, where $\mathbf{D}$ is the degree matrix with $D_{ii} = \sum_j \mathcal{S}_{ij}$. Then, we can develop the numerator term of $\eta_r$ in Eq. (9) as follows:

$$\begin{aligned} T_1 &= \sum_{i,j} \left( f_{ri} - f_{rj} \right)^2 \mathcal{S}_{ij} \\ &= \sum_{i,j} \left( f^2_{ri} - 2 f_{ri} f_{rj} + f^2_{rj} \right) \mathcal{S}_{ij} \\ &= 2 \left( \sum_{i,j} f^2_{ri} \mathcal{S}_{ij} - \sum_{i,j} f_{ri} \mathcal{S}_{ij} f_{rj} \right) \\ &= 2 \left( \mathbf{f}_r^{\mathrm{T}} \mathbf{D} \mathbf{f}_r - \mathbf{f}_r^{\mathrm{T}} S \mathbf{f} \right) \\ &= 2 \mathbf{f}_r^{\mathrm{T}} \mathbf{L} \mathbf{f}_r \end{aligned} \tag{15}$$

The global structure is modeled by variance $\Sigma_r$ of the whole dataset X, and both inner-class variance $\Sigma^w_r$ and inter-class variance $\Sigma^b_r$ of the labeled dataset $\mathrm{X}^l$. To compute $\Sigma_r$, define $1 = [1, 1, \cdots, 1]$, let

$$\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^{\mathrm{T}} \mathbf{D} 1}{1^{\mathrm{T}} \mathbf{D} 1} 1 \tag{16}$$

And according to [18],

$$\begin{aligned}\Sigma_r &= \sum_i (f_{ri}-\mu_r)^2 D_{ii} \\ &= \sum_i \left(f_{ri}-\sum_j f_{rj}\frac{D_{jj}}{\sum_k D_{kk}}\right)^2 D_{ii} \\ &= \sum_i \left(f_{ri}-\frac{1}{\sum_k D_{kk}}\sum_j f_{rj} D_{jj}\right)^2 D_{ii} \\ &= \sum_i \left(f_{ri}-\frac{\mathbf{f}_r^{\mathrm{T}}\mathbf{D}1}{1^{\mathrm{T}}\mathbf{D}1}\right)^2 D_{ii} \\ &= \sum_i \tilde{f}_{ri}^2 D_{ii} \\ &= \tilde{\mathbf{f}}_r^{\mathrm{T}}\mathbf{D}\tilde{\mathbf{f}}_r\end{aligned} \tag{17}$$

To compute $\Sigma_r^w$ and $\Sigma_r^b$, a similarity matrix $S^l$ is defined, whose elements are as follows:

$$\mathcal{S}_{ij}^l = \begin{cases} 1/n_c & y_i = y_j = c \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

To simplify, we assume the instances are ordered according to their labels and the unlabeled data points are appended after the labeled ones. Thus, $S^l$ can be written as follows:

$$S^l = \begin{bmatrix} S_1^l & 0 & 0 & \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & S_C^l & \\ & 0 & & 0 \end{bmatrix} \tag{19}$$

where, $S_c^l$ is a $n_c \times n_c$ matrix whose elements are $1/n_c$ and 0 is a matrix whose elements are all zero.

And define the Laplacian matrix as $\mathbf{L}^l = \mathbf{D}^l - S^l$, where $\mathbf{D}^l$ are the degree matrix with $D_{ii}^l = \sum_j \mathcal{S}_{ij}^l$. Note that for each $S_c^l$, the raw sum is equal to 1, so

$$\mathbf{D}^l = \begin{bmatrix} \mathbf{D}_1^l & 0 & 0 & \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \mathbf{D}_C^l & \\ & 0 & & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_L & 0 \\ 0 & 0 \end{bmatrix} \tag{20}$$

where, $\mathbf{I}_L$ is a $L \times L$ identity matrix in which $L$ is the number of the labeled data as given in the Section 2, and $\mathbf{D}_c^l$ is a $n_c \times n_c$ identity matrix.

Thus, the inner-class covariance $\Sigma_r^w$ can be developed as follows:

$$\begin{aligned}\Sigma_r^w &= \sum_c n_c\left(\sigma_r^{(c)}\right)^2 \\ &= \sum_c n_c \mathrm{cov}\left(\mathbf{f}_r^{(c)}, \mathbf{f}_r^{(c)}\right) \\ &= \sum_c \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} \left(\mathbf{D}_c^l - S_c^l\right) \mathbf{f}_r^{(c)} \\ &= \mathbf{f}_r^{\mathrm{T}} \mathbf{L}^l \mathbf{f}_r = \left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{L}^l \tilde{\mathbf{f}}_r^l\end{aligned} \tag{21}$$

where, $\mathbf{f}_r^{(c)}$ is an $N \times 1$ vector whose elements are as follows:

$$f_{ri}^{(c)} = \begin{cases} f_{ri} & \text{if } y_i = c \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

and

$$\tilde{\mathbf{f}}_r^l = \mathbf{f}_r - \frac{\mathbf{f}_r^{\mathrm{T}} \mathbf{D}^l 1}{1^{\mathrm{T}} \mathbf{D}^l 1} 1 \tag{23}$$

and the inter-class covariance $\Sigma_r^b$ can be develop as follows:

$$\begin{aligned}\Sigma_r^b &= \sum_c n_c \left(\mu_r^{(c)} - \mu_r^l\right)^2 \\ &= \sum_c \left(n_c \left(\mu_r^{(c)}\right)^2 - 2 n_c \mu_r^{(c)} \mu_r^l + n_c \left(\mu_r^l\right)^2\right) \\ &= \sum_c \frac{1}{n_c} \left(n_c \mu_r^{(c)}\right)^2 - 2\mu_r^l \sum_c n_c \mu_r^{(c)} + \left(\mu_r^l\right)^2 \sum_c n_c \\ &= \sum_c \frac{1}{n_c} \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} 1 1^{\mathrm{T}} \mathbf{f}_r^{(c)} - 2L\mu_r^l + L\left(\mu_r^l\right)^2 \\ &= \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} S^l \mathbf{f}_r^{(c)} - \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} \left(\frac{1}{L} 1 1^{\mathrm{T}}\right) \mathbf{f}_r^{(c)} \\ &= \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} \left(\mathbf{D}^l - S^l\right) \mathbf{f}_r^{(c)} - \left(\mathbf{f}_r^{(c)}\right)^{\mathrm{T}} \left(\mathbf{D}^l - \frac{1}{L} 1 1^{\mathrm{T}}\right) \mathbf{f}_r^{(c)} \\ &= \left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{L}^l \tilde{\mathbf{f}}_r^l - \left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{D}^l \tilde{\mathbf{f}}_r^l\end{aligned} \tag{24}$$

Thus,

$$T_2 = \Sigma_r + \Sigma_r^b - \Sigma_r^w = \tilde{\mathbf{f}}_r^{\mathrm{T}} \mathbf{D} \tilde{\mathbf{f}}_r + 2\left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{L}^l \tilde{\mathbf{f}}_r^l - \left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{D}^l \tilde{\mathbf{f}}_r^l \tag{25}$$

Subsequently, the CCLS can be computed as follows:

$$\eta_r = \frac{2\mathbf{f}_r^{\mathrm{T}} \mathbf{L} \mathbf{f}_r}{\tilde{\mathbf{f}}_r^{\mathrm{T}} \mathbf{D} \tilde{\mathbf{f}}_r + 2\left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{L}^l \tilde{\mathbf{f}}_r^l - \left(\tilde{\mathbf{f}}_r^l\right)^{\mathrm{T}} \mathbf{D}^l \tilde{\mathbf{f}}_r^l} \tag{26}$$

**Algorithm 1**: CCLS

**Input**: Data set $\mathrm{X}$, parameters $\lambda$ and $\gamma$

**Output**: The ranked feature list

1: Construct the constraint set $\Omega_{ML}$ and $\Omega_{CL}$ from $\mathrm{X}^l$ and $\mathrm{Y}^l$;

2: Construct graphs $\boldsymbol{G}$, $\boldsymbol{G}^M$, and $\boldsymbol{G}^C$ from $\mathrm{X}$, $\Omega_{ML}$, and $\Omega_{CL}$, respectively;

3: Calculate the similarity matrices $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}}^l$, the degree matrices $\boldsymbol{D}$ and $\boldsymbol{D}^l$, and the Laplacian matrices $\boldsymbol{L}$ and $\boldsymbol{L}^l$;

4: **for** $r=1$ **to** $M$ **do**

5: Calculate $\eta_r$ as Eq. (26);

6: **end for**

7: Rank the features $F_r$ according to their $\eta_r$ in ascending order.

The whole procedure of the proposed CCLS is summarized in Algorithm 1. Now we analyze the time complexity of Algorithm 1. Step 1 constructs the constraint set requiring $O(L^2)$ operations. Step 2–3 build the graph matrices requiring $O(N^2)$ operations. Step 4–6 evaluate the $M$ features based on graphs, requiring $O(MN^2)$ operations. Step 7 ranks the $M$ features according to their scores requiring $O(M\log M)$ operations. Thus, the overall time complexity of Algorithm **1** is $O(M\times \max(N^2, \log M))$.

## 4.2 Feature redundancy

In this section, to evaluate the effectiveness of features, some measurements of redundancy between features are introduced firstly. Then, our strategy to measure similarity between features is given.

### *4.2.1 Measurements based on MI or correlation*

Redundancy is usually characterized in terms of mutual information or correlation, in which the former one is the most widely used measure matric. MI is defined as Eq. (6), and as mentioned above, the MI is difficult to compute when at least one of the features is continuous though there have been many researches [19, 28] focusing on solving this problem.

If two features have a strong correlation between their values, it can be sure that they are redundant to each other. Thus, it is natural to use feature correlation to measure redundancy. Among the kinds of correlation coefficients, Pearson correlation coefficient is the most widely used measure. For two features $F_r$ and $F_v$, the Pearson correlation coefficient between them is defined as follows:

$$r(F_r, F_v) = \frac{\sum_i (f_{ri}-\mu_r)(f_{vi}-\mu_v)}{\sqrt{\sum_i (f_{ri}-\mu_r)^2}\sqrt{\sum_i (f_{vi}-\mu_v)^2}} \tag{27}$$

The large $|r(F_r, F_v)|$ means high correlation and redundancy. However, this coefficient can only measure the linear correlation properties, which may cause more errors when the relationship between features is non-linear.

#### 4.2.2 GMM-based Bhattacharyya distance

Besides MI and correlation, redundancy can also be measured by the distance function. For a feature $F_r$ can be regarded as a random variable, it's easy to extract a probabilistic distance from some parameters of the corresponding feature vector $\mathbf{f}_r$ with the assumption of underlying distributions. It has been observed that Bhattacharyya distance is more effective than other distance functions like Euclidean, Kullback-Leibler, and Fisher [3, 20]. Moreover, The Bhattacharyya distance [13] has been used as a distance measure of vectors in feature extraction [6] and feature selection [26, 32]. Here we focus on Bhattacharyya-distance-based redundancy measurement.

In its simplest formulation, the Bhattacharyya distance between two Gaussian distributions $g_r \sim \mathcal{N}(\mathbf{m}_r, \mathbf{\Sigma}_r)$ and $g_v \sim \mathcal{N}(\mathbf{m}_v, \mathbf{\Sigma}_v)$ is defined as follows:

$$D_B(g_r, g_v) = \frac{1}{8}(\mathbf{m}_r - \mathbf{m}_v)^{\mathrm{T}} \left[\frac{\mathbf{\Sigma}_r + \mathbf{\Sigma}_v}{2}\right]^{-1} (\mathbf{m}_r - \mathbf{m}_v) + \frac{1}{2}\ln \frac{\left|\frac{\mathbf{\Sigma}_r + \mathbf{\Sigma}_v}{2}\right|}{\sqrt{|\mathbf{\Sigma}_r||\mathbf{\Sigma}_v|}} \tag{28}$$

where, $\mathcal{N}(\mathbf{m}_r, \mathbf{\Sigma}_r)$ represents a multi-dimensional Gaussian distributions with mean vector $\mathbf{m}_r$ and covariance matrix $\mathbf{\Sigma}_r$:

$$\mathcal{N}(\mathbf{m}_r, \mathbf{\Sigma}_r) = \frac{1}{(2\pi)^{M'/2}|\mathbf{\Sigma}_r|^{1/2}} \exp\left\{-\frac{(\mathbf{o}_r - \mathbf{m}_r)^{\mathrm{T}} \mathbf{\Sigma}_r^{-1} (\mathbf{o}_r - \mathbf{m}_r)}{2}\right\} \tag{29}$$

and $\mathbf{o}_r$ is the random variable and $M'$ is its dimension. So does $\mathcal{N}(\mathbf{m}_v, \mathbf{\Sigma}_v)$.

The feature $F_r$ (or $F_v$) is naturally treated as a random variable which follows single Gaussian distribution which mean ($\mathbf{m}_r$ or $\mathbf{m}_v$) and variance ($\mathbf{\Sigma}_r$ or $\mathbf{\Sigma}_v$) can be estimated from $\mathbf{f}_r$ (or $\mathbf{f}_v$). Then, the Bhattacharyya distance $D_B(g_r, g_v)$ can be used to measure the redundancy between $F_r$ and $F_v$.

However, it remains in doubt whether it is suitable to approximate the distribution of $F_r$ by single Gaussian, because $F_r$ contains data of at least two classes. Thus, a GMM-based Bhattacharyya distance to measure the feature redundancy is proposed, which can be stated as follows:

(1) The unlabeled data in $\mathrm{X}^u$ is classified by using the nearest neighborhood (1-NN) classifier based on the labeled data in $\mathrm{X}^l$. Then, the "labels" of the unlabeled data is $\mathrm{Y}^u = \{y_{L+1}, y_{L+2}, \cdots, y_N | y_i = 1, 2, \cdots, C\}$.

(2) For the $r$-th feature $F_r$, we normalize its feature vector $\mathbf{f}_r$ to be a new vector $\mathbf{f}_r'$ with zero mean and unit variance:

$$\mathbf{f}_r' = \frac{\mathbf{f}_r - \mu_r 1}{\sigma_r} \tag{30}$$

(3) Suppose $F_r'^{(c)}$ is the $r$-th normalized feature of class $c$, we use a GMM estimated from $\mathbf{f}_r'^{(c)} = \left\{f_{ri}' | y_i = c\right\}$ to approximate its distribution:

$$F_r'^{(c)} \sim \sum_{k=1}^{K^{(c)}} \omega_{r,k}^{(c)} g_{r,k}^{(c)} \tag{31}$$

where, $K^{(c)}$ is the number of Gaussians in the GMM for class $c$ which is determined according to the number of instances in $\mathbf{f}_r'^{(c)}$. $\omega_{r,k}^{(c)}$ is the weight of the $k$-th mixture

component, and $g_{r,k}^{(c)} \sim \mathcal{N}\left(m_{r,k}^{(c)}, \Sigma_{r,k}^{(c)}\right)$ is the Gaussian distribution of the $k$-th mixture component. Thus, the distribution of the $r$-th normalized feature $F_r^{'}$ is:

$$F_r^{'} \sim \sum_c \frac{1}{C} \sum_{k=1}^{K^{(c)}} \omega_{r,k}^{(c)} g_{r,k}^{(c)} \tag{32}$$

(4) For any feature pairs $(F_r, F_v)$, the redundancy between them is defined as follows:

$$\theta_{rv} = \sum_c \frac{1}{C} \sum_{k=1}^{K^{(c)}} \sum_{\kappa=1}^{K^{(c)}} \omega_{r,k}^{(c)} \omega_{r,\kappa}^{(c)} D_B\left(g_{r,k}^{(c)}, g_{v,\kappa}^{(c)}\right) \tag{33}$$

Define the redundancy matrix $\boldsymbol{\Theta}$ whose element $\Theta_{rv}$ is as follows:

$$\Theta_{rv} = \begin{cases} \theta_{rv} & r \neq v \\ 0 & r = v \end{cases} \tag{34}$$

### 4.3 The complete framework of SSMRMR

In this section, we will give a view of the complete framework of SSMRMR. Moreover, we will present the incremental search which obtains a near-optimal solution efficiently.

#### *4.3.1 The objective function*

Similar to mRMR, SSMRMR obtains the optimal feature set by maximum feature relevance and minimum feature redundancy simultaneously. Thus, the objective function of SSMRMR is defined as follows:

$$\min_{\Lambda \subset \Omega} \left\{ \sum_{\mathbf{f}_r \in \Lambda} \eta_r^{'} - \frac{1}{|\Lambda|} \sum_{\mathbf{f}_r, \mathbf{f}_v \in \Lambda} \Theta_{rv}^{'} \right\} \tag{35}$$

where, $\eta_r^{'} = \eta_r / \max(\boldsymbol{\Psi})$ and $\Theta_{rv}^{'} = \Theta_{rv} / \max(\boldsymbol{\Theta})$. The normalization of these measurements is done to reduce the effect of differences in magnitude between feature relevance and redundancy.

#### *4.3.2 Incremental search algorithm*

The time complexity is $O(M^{|\Lambda|})$ when the exact solution of the optimization in Eq. (35) is obtained. However, the incremental search algorithm can be used which obtains the near-optimal features with $O(M \cdot |\Lambda|)$ search. The algorithm steps are as follows:

(1) The feature with the minimum constraint compensated Laplacian score is obtained as the first optimal feature:

$$\mathbf{f}_1 = \underset{\mathbf{f}_r \subset \Omega}{\operatorname{argmin}} \, \eta_r \tag{36}$$

and $\Lambda_1 = \{\mathbf{f}_1\}$.

(2) To select the $m$-th feature, the corresponding incremental algorithm optimizes the following condition:

$$\mathbf{f}_m = \underset{\mathbf{f}_r \subset \Omega - \Lambda_{m-1}}{\operatorname{argmin}} \left\{ \eta'_r - \frac{1}{m-1} \sum_{\mathbf{f}_v \in \Lambda_{m-1}} \Theta'_{rv} \right\} \tag{37}$$

where, $\Lambda_{m-1}$ is the optimal feature set with $m-1$ features.

(3) Iterate step 2 until an expected feature number $R$ have been obtained.

#### *4.3.3 Complete framework*

The whole procedure of SSMRMR algorithm is summarized in Algorithm 2. The relevancy vector and redundancy matrix are computed by using CCLS algorithm and GMM-based Bhattacharyya distance. Then, the optimal feature set is selected by using first-order incremental search algorithm.

**Algorithm 2**: SSMRMR

**Input**: Data set $\mathrm{X}$, expected number $R$, parameters $\lambda$ and $\gamma$

**Output**: The optimal feature set

1: Compute the feature relevancy vector $\boldsymbol{\Psi}$ as Algorithm 1;

2: Compute the feature redundancy matrix $\boldsymbol{\Theta}$ as Eq. (34);

3: Obtain the first optimal feature $\hat{\boldsymbol{f}}_1$ as Eq. (36);

4: **for** $m = 2$ **to** $R$ **do**

5: Select the $m$-th optimal feature $\hat{\boldsymbol{f}}_m$ as Eq. (37);

6: **end for**

7: Return the optimal feature set $\Lambda$.

## 5 Experimental study

In this Section, we firstly illustrate several features which have been widely used in audio classification. Then we evaluate the performance of relevance and redundancy measurements respectively. At last, we test our feature selection approach in audio classification.

### 5.1 Audio classification

Audio segmentation is a type of methods which split an audio stream into segments of homogeneous content. Given a predefined set of audio classes, some methods segment audios by executing iterative steps of segmentation and classification jointly, which means classification is embedded in audio segmentation in these methods. Assuming that an audio signal has been divided into a sequence of audio segments using fixed window segmentation, our works focus on categorizing these audio segments into a set of predefined audio classes. Although

there may be some differences between the traditional definition of audio classification and that in our work, the essential issues are the same.

Figure 1 illustrates the process of audio classification. In an audio classification system, every audio signal is first divided into mid-length segments which duration range from 0.5 to 10 s. After this, the selected features are extracted for each segment using short-term overlapping frames. The sequence of short-term features in each segment is used to compute feature statistics, which are used as inputs to the classifier. In the final classification stage, the classifier determines a segment-by-segment decision.

In audio analysis and classification there are dozens of features which can be used. Moreover, many novel feature extraction methods are proposed constantly [14, 25, 40]. In this paper, some classical and widely used acoustic features are selected for feature selection sources. Widely-used time-domain features [15] include short-term energy [22], zero-crossing rate [29], and entropy of energy [16]. Common frequency-domain features include spectral centroid, spectral spread, spectral entropy [21], spectral flux, spectral roll-off, MFCCs, and chroma vector [1].

In our system, the audio segment has been divided into 500 ms sub-segments without overlapping. And then, these sub-segments are split into overlapped 32 ms short-term frames with 10 ms frame shift, resulting 50 frames for each sub-segment. The 35 dimensional short-term feature vectors (shown in Table 1) are extracted from short-term frames. For each sub-segment, the mean and standard deviation of the corresponding 50 short-term feature vectors are computed and concatenated together, resulting 70 dimensional mid-term feature vectors which are used for classification.

### 5.2 Data and experimental setup

Experiments were performed using audio signals under telephone channel. Thus, each audio segment may contain speech, non-speech or silence, with more detailed classes as shown in Fig. 2. ‘Speech’ indicates direct dialogues between the calling and called users, when the call is connected, while ‘silence’ implies the segment with comfort noise. ‘Non-speech’ can be sub-classified into four types: ring, music, song, and other. ‘Ring’ contains the single-tone, dual-

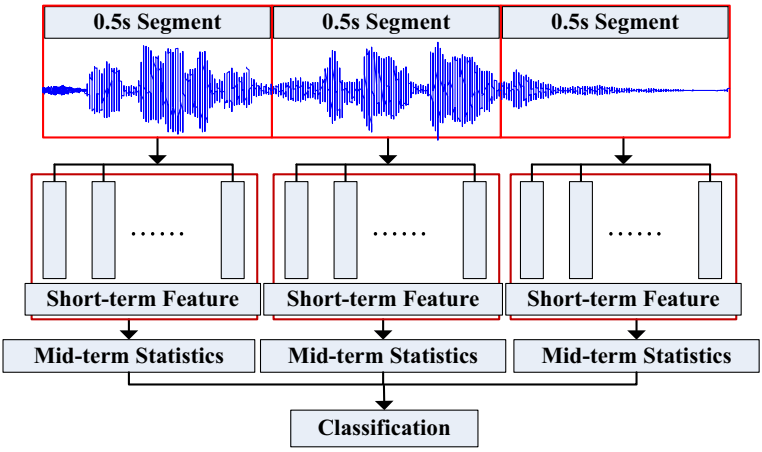


**Fig. 1** The audio classification framework

**Table 1** Classification accuracy of different features

| Short-term Feature | | Mid-term Feature | | Accuracy |
|---|---|---|---|---|
| Types | Dimension | Types | Dimension | |
| Zero-crossing Rate | 1 | Mean | 1 | 73.73 |
| | | STD | 1 | 74.86 |
| | | Mean and STD | 2 | 75.10 |
| Short-term Energy | 1 | Mean | 1 | 45.81 |
| | | STD | 1 | 46.03 |
| | | Mean and STD | 2 | 69.41 |
| Energy Entropy | 1 | Mean | 1 | 71.86 |
| | | STD | 1 | 69.10 |
| | | Mean and STD | 2 | 74.99 |
| Spectral Centroid | 2 | Mean | 2 | 79.19 |
| | | STD | 2 | 74.49 |
| | | Mean and STD | 4 | 84.79 |
| Spectral Entropy | 1 | Mean | 1 | 69.33 |
| | | STD | 1 | 74.27 |
| | | Mean and STD | 2 | 76.86 |
| Spectral Flux | 1 | Mean | 1 | 79.21 |
| | | STD | 1 | 69.09 |
| | | Mean and STD | 2 | 77.86 |
| Spectral Roll-off | 1 | Mean | 1 | 71.80 |
| | | STD | 1 | 74.20 |
| | | Mean and STD | 2 | 74.13 |
| MFCCs | 13 | Mean | 13 | 84.26 |
| | | STD | 13 | 86.44 |
| | | Mean and STD | 26 | **87.66** |
| Harmonic | 2 | Mean | 2 | 69.99 |
| | | STD | 2 | 82.90 |
| | | Mean and STD | 4 | 83.13 |
| Chroma Vector | 12 | Mean | 12 | 83.49 |
| | | STD | 12 | 83.87 |
| | | Mean and STD | 24 | 83.73 |
| All | 35 | Mean | 35 | 81.07 |
| | | STD | 35 | 85.97 |
| | | Mean and STD | 70 | 86.04 |

tone, or multi-tone used for dialing or waiting warning. ‘music’ and ‘song’ are the waiting music before the call is connected or the environmental noise when the phone is in call. ‘Other’

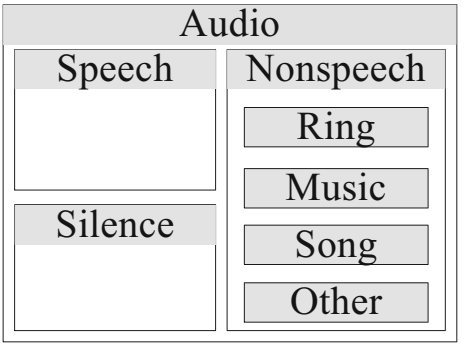


**Fig. 2** The audio classes in telephone channel

**Table 2** Averaged accuracy of different algorithms. (400 labeled segments)

| Alg. | Spec | ReliefF | LS | CS | CLS | CCLS |
|---|---|---|---|---|---|---|
| Ave. | 85.26 ± 4.66 | 86.41 ± 2.79 | 85.32 ± 3.07 | 84.62 ± 2.92 | 83.40 ± 4.56 | 88.46 ± 3.67 |
| Opt. | 89.97 | 90.90 | 89.08 | 88.95 | 88.27 | 91.14 |
| Num. | 23 | 19 | 33 | 39 | 47 | 26 |

includes special sounds, such as laugh, barking, coughing, or other isolated sounds. To be specific, the mixed segments (speech over music) are not allowed.

The database used here has been collected and manually labeled by Tsinghua University. It contains about 7 h audios with 837 real telephone recordings. The speaker in each recording is different, so does the waiting music. And the corpus consists of 204.4 min 'speech' data, 12.7 min 'ring' data, 6.3 min 'music' data, 6.6 min 'song' data, and 1.2 min 'other' data.

According to the label, an audio signal, which contains speech or non-speech, is divided into several 0.5 s segments. For each segment, all features mentioned in section 2 are extracted based on the short-term analysis, and the dimension of short-term feature is 35. The frame length and frame shift size are 32 ms and 10 ms, respectively. Then the two mid-term statistics, mean and standard deviation, are drawn per feature, therefore, the dimension of mid-term statistics vector is 70.

For feature selection, we choose 2000 speech segments and 2000 non-speech segments, with only 400 randomly chosen labeled segments. The $\gamma$ value is set to 0.9 and $\lambda = 0.5$. We compare CCLS with existing unsupervised Laplacian Score, as well as supervised Constraint Score, Constrained Laplacian Score (CLS) [2], spectral feature selection (Spec) [38], and ReliefF [27]. The GMM-based Bhattacharyya distance is compared with MI-based and correlation-based measurements. We use a development dataset containing 200 speech segments and 200 non-speech segments to choose the optimal feature subset. And in the test dataset, there are 500 speech segments and 500 non-speech segments.

In all experiments, the k-nearest neighborhood (KNN) classifier with Euclidean distance is utilized for classification after feature selection and $k = 5$. To avoid the influence of the classifier, the training datasets of the classifier for all experiments are kept the same.

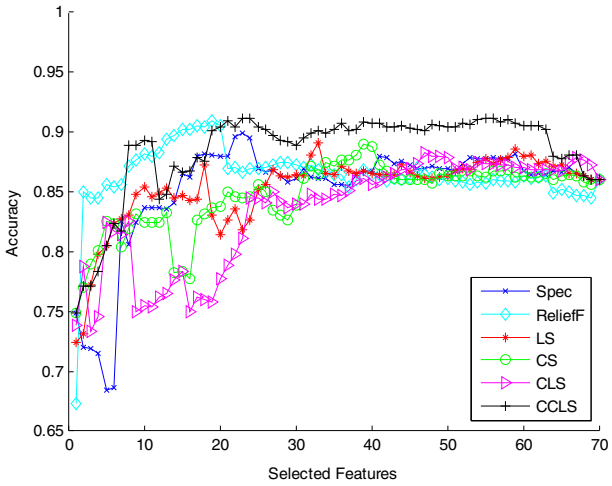


**Fig. 3** Accuracy vs. different numbers of selected features

**Table 3** Performance of supervised and semisupervised methods with 200 labeled segments

| Algorithms | Spec | ReliefF | CS | CCLS |
|---|---|---|---|---|
| Ave. | 83.33 ± 3.94 | 85.53 ± 5.92 | 81.06 ± 3.34 | 87.62 ± 3.50 |
| Opt. | 87.20 | 89.24 | 87.68 | 91.64 |
| Num. | 57 | 35 | 55 | 40 |

We use accuracy (*Acc*), average accuracy (*Ave*), optimized accuracy (*Opt*), and optimized number of feature (*Num*) to evaluate the performance of algorithms. The definitions are as follows:

$$Acc = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{38}$$

where, $N_{\text{correct}}$ is the number of segments which are classified correctly, and $N_{\text{total}}$ is the total number of both speech segments and non-speech segments, namely $N_{\text{total}} = 1000$.

$$Ave = \frac{1}{M} \sum_{R=1}^{M} \text{Acc}(R) \tag{39}$$

where, the mid-term feature dimension is $M = 70$, $R$ is the number of selected features, and $Acc(R)$ is the accuracy when using the selected $R$ features for classification.

$$Opt = \max \text{Acc}(R) \tag{40}$$

and

$$Num = \arg\max_{R} \text{Acc}(R) \tag{41}$$

The *Num* measurement is used to evaluate the redundancy of selected feature sets.

### 5.3 Experimental results

Ten types of short-term features extracted are listed in Table 1. Two statistics, mean and standard deviation (STD), are used as the mid-term representation of the audio segments. Table 1 shows the classification accuracies of different features for audio classification. The top 3 best features are MFCCs, chroma vector, and spectral centroid and the worst feature is short-term energy. Moreover, using all of these features does not improve but rather decreases the accuracy, as seen by comparing results using MFCC with that of all features, which indicates that there is redundant and even contradictory information among the features. Thus, it is valuable to use feature selection as a preprocessing module.

**Table 4** Performances of supervised and semisupervised methods with 800 labeled segments

| Algorithms | Spec | ReliefF | CS | CCLS |
|---|---|---|---|---|
| Ave. | 86.76 ± 2.82 | 88.47 ± 4.69 | 87.24 ± 3.48 | 89.72 ± 3.71 |
| Opt. | 89.49 | 92.27 | 91.45 | 92.71 |
| Num. | 55 | 18 | 23 | 25 |

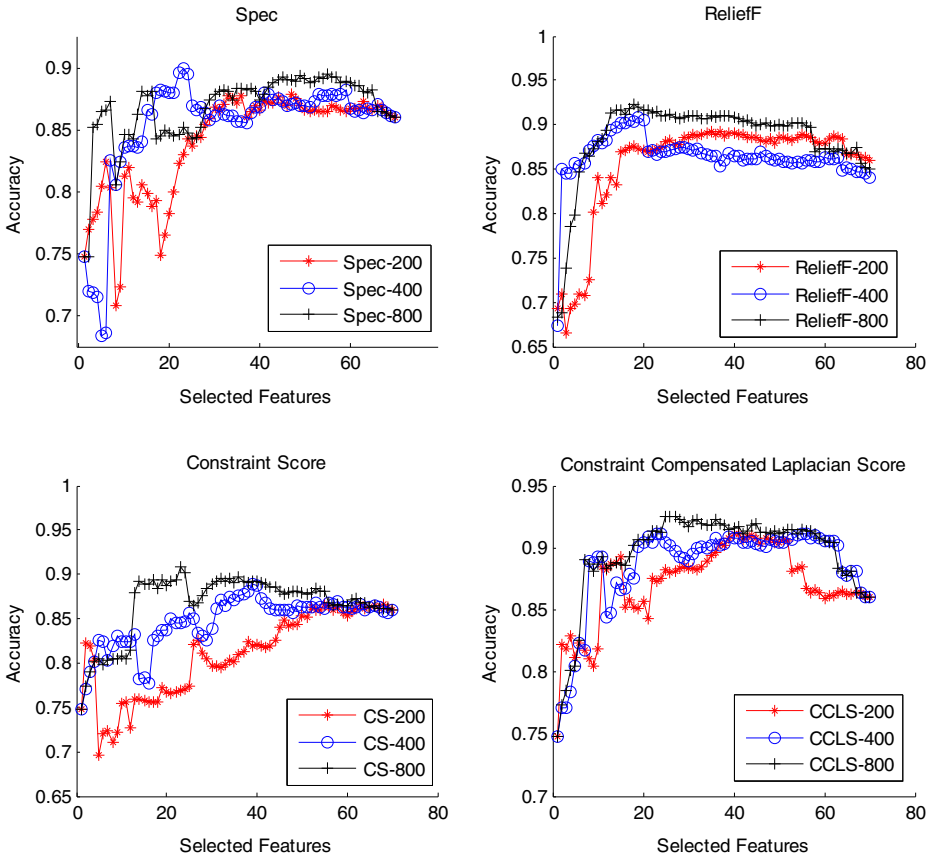


**Fig. 4** Accuracy vs. different numbers of selected features and different numbers of labeled data segments

### *5.3.1 Feature relevance*

To further illustrate the effectiveness of CCLS, it is compared with several established feature selection methods, which include Spec, ReliefF, LS, CS and CLS.

Table 2 gives comparisons of the averaged accuracy, optimized accuracy and the optimized number of features. In addition the value after the symbol '±' denotes the standard deviation. It indicates that the performance is significantly improved by using the first $d$ features selected from the ranking list of features generated by feature selection algorithms. It means that there is redundant and even contradictory information among original feature space, and feature selection algorithm can remove irrelevant and redundant features effectively.

The CCLS is superior to other evaluated methods not only in terms of averaged accuracy but also in terms of optimized accuracy. On the other hand, the CLS has the lowest averaged accuracy and optimized accuracy.

Figure 3 shows accuracy vs. number of selected features. It can be seen that the performance of CCLS is significantly better than that of Spec, Laplacian Score, Constraint Score and Constrained Laplacian Score. The results supports that combining supervised information with data structures to evaluate the relevance of features is very useful in feature selection.

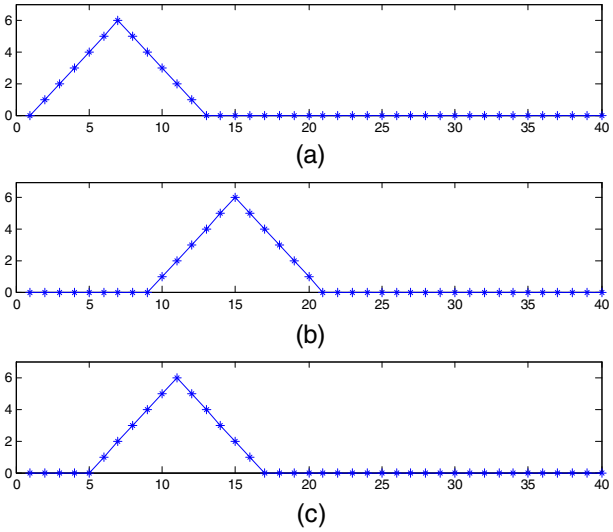


**Fig. 5** The base waveforms of waveform database generator data set

To explore the influence of the numbers of labeled segments on the performance of the algorithm, different numbers of labeled data are used. The averaged accuracies, optimized accuracies and the optimal numbers of features of such methods on the condition of 200 and 800 labeled segments are summarized in Table 3 and Table 4 respectively. Comparing Table 2 with Tables 3 and 4, it is easy to conclude that the performance improves when the number of labeled data segments increases from 200 to 800. The CCLS is best in terms of averaged accuracy and optimized accuracy regardless

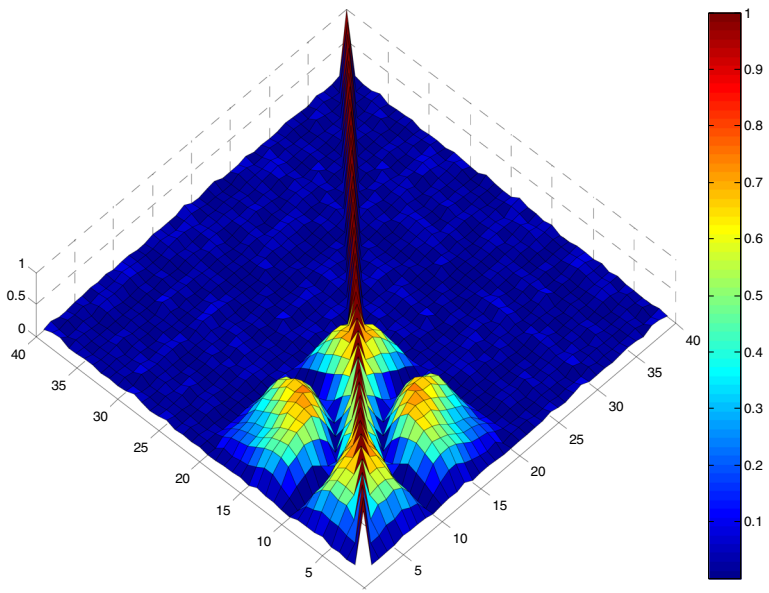

**Fig. 6** The feature redundancy based on Pearson correlation coefficient

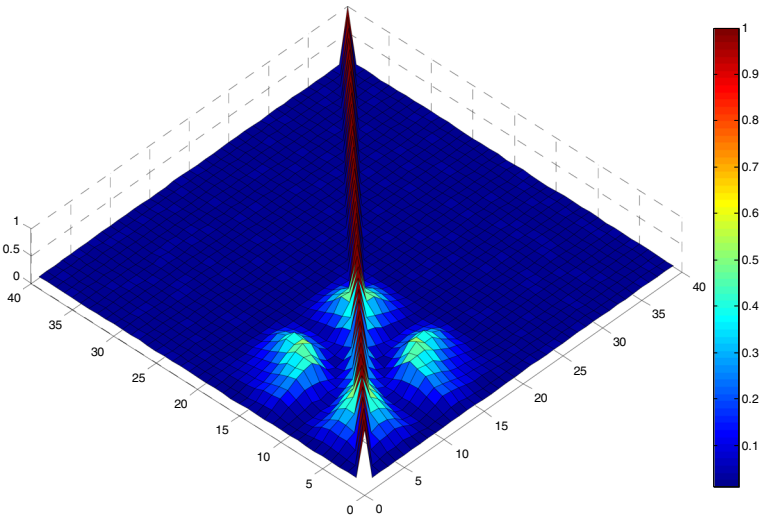


**Fig. 7** The feature redundancy based on mutual information. The mutual information is computed using Gaussian kernel-based estimator [31]

of the number of labeled segments. The optimal feature number of ReliefF is always smaller than others. This may indicate that there are some redundant features in the optimum feature set selected by CCLS method.

Figure 4 shows the plots of accuracy vs. the number of selected features and the amount of labeled data. However, it should also be noticed that the performances of CCLS and ReliefF do not drop rapidly when decreasing the number of labeled data to 200, while the CS and Spec algorithms are unable to select relevant features.

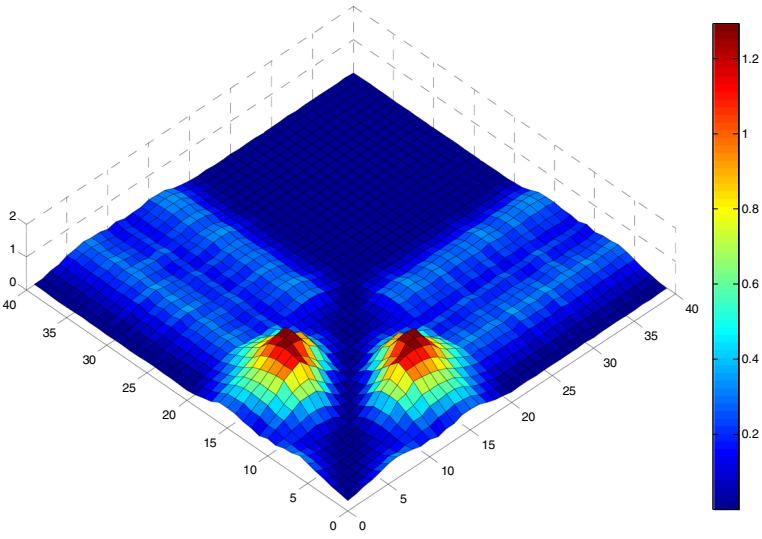


**Fig. 8** The feature redundancy measured by GMM-based Bhattacharyya distance. Note that the redundancy is inversely proportional to this measurement

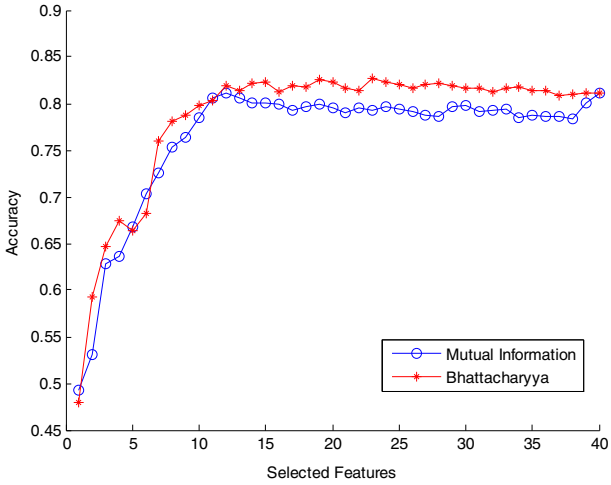


**Fig. 9** Accuracy vs. different numbers of selected features using mRMR feature selection algorithm with different redundancy measurement

### 5.3.2 Feature redundancy

To examine the effectiveness of GMM-based Bhattacharyya distance in measuring feature redundancy, some experiments have been done on the Waveform Database Generator (Version 2) Data Set [4]. This data set contains 5000 40-dimensional instances from 3 classes. Each class is generated from a combination of two of three base waveforms, $h_1(t)$, $h_2(t)$, and $h_3(t)$. Figure 5 shows graphs of these base waveforms.

To generate a instance $\mathbf{x}_i$, a single uniform random number $u \sim U(0,1)$ and 40 normal random numbers $e_t \sim \mathcal{N}(0,\sigma^2)$, $t = 1, 2, \cdots, 40$ are generated. Then, $\mathbf{x}_i$ is generated by combining two of the three base waveforms as follow:

$$\mathbf{x}_i = u\mathbf{h}_1 + (1-u)\mathbf{h}_2 + \mathbf{e} \tag{42}$$

where, for class 1, $\mathbf{h}_1 = [h_1(t)]$ and $\mathbf{h}_2 = [h_2(t)]$. For class 2, $\mathbf{h}_1 = [h_1(t)]$ and $\mathbf{h}_2 = [h_3(t)]$ are selected to generate instances. For class 3, $\mathbf{h}_1 = [h_2(t)]$ and $\mathbf{h}_2 = [h_3(t)]$ are used similarly. And $\mathbf{e} = [e_t | t = 1, 2, \cdots, 40]^{\mathrm{T}}$. In all cases, there are many irrelevant features, almost half of them, which can be removed to achieve the best performance. This not only improves classification accuracy, but also reduces the time complexity of classification.

Obviously, the features $\mathbf{f}_1, \mathbf{f}_{21}, \mathbf{f}_{22}, \cdots, \mathbf{f}_{40}$ in this data set are white noise features for all of the corresponding values of base waveforms are 0. They are uncorrelated to each other or other relevant features. In other words, the redundancy corresponding to noise feature is quite low while using Pearson correlation coefficient as the measurement, shown in Fig. 6.

Similarly, the mutual information-based measurement faces the same defect as Pearson correlation, shown in Fig. 7.

**Table 5** The Number of Noise Feature Among the First 19 Features Selected by the mRMR Algorithm

| Redundancy | MI | Bhattacharyya |
|---|---|---|
| Number | 2 ($f_1$ and $f_{30}$) | 0 |

**Table 6** Performance Comparison of mRMR, CCLS, and SSMRMR Algorithm(400 labeled segments)

| Algorithms | mRMR | CCLS | SSMRMR |
|---|---|---|---|
| Ave. | 85.93 ± 3.70 | 88.46 ± 3.67 | 88.63 ± 3.55 |
| Opt. | 90.09 | 91.14 | 91.65 |
| Num. | 22 | 26 | 18 |

Figure 8 shows the graph of GMM-based Bhattacharyya distance measurement. In this graph, the larger value means smaller redundancy. It is easy to see that the redundancy related to noise feature is large enough, so it will be helpful for noise feature removal. Moreover, comparing Fig. 8 with Figs. 6 and 7, it's not difficult to find the results in the region with vertices at about (6, 13), (6, 16), (9, 13), (9, 16) are quite the contrary. Figure 6 and Fig. 7 show that the redundancy values among these features are the highest while Fig. 8 indicates that these redundancy values are lowest. It is mainly because that the correlation among these features caused by the random variable is non-linear which cannot be represented properly by correlation-based or MI-based measurements. And the redundancy measurement proposed in this paper can prevent this type of problem.

We randomly choose 100 instances for each class from this dataset as training data. The MI between the feature and targeted classes is estimated using the nearest-neighbor method [28] and the MI between features is computed by Gaussian kernel-based estimator [31]. The GMM-based Bhattacharyya distance is calculated without Step 1 for all instances are labeled data.

Figure 9 shows the plots of mRMR algorithm with different redundancy measurements for accuracy vs. different numbers of selected features. It can be seen that the performance with GMM-based Bhattacharyya distance measurement is better than that with mutual information-based measurement. Table 5 shows the number of noise feature among the first 19 features. It can be concluded from these experimental results that mutual information-based measurement cannot represent the redundancy properly when the data set is affected by the random variable, which leads the algorithm to tend to preferentially choose the noise feature.

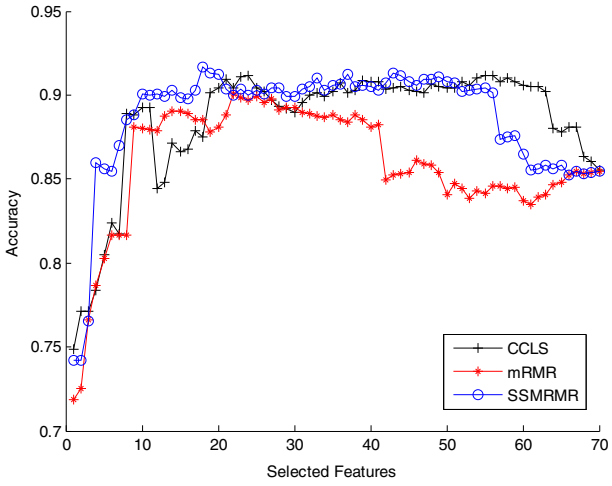


**Fig. 10** Accuracy vs. different numbers of selected features using mRMR, CCLS, and SSMRMR feature selection algorithm

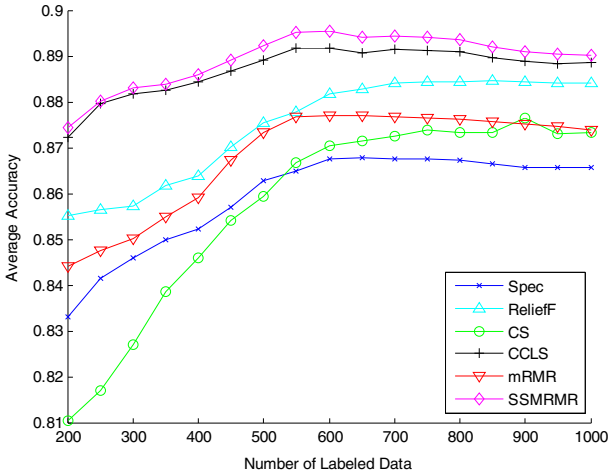


**Fig. 11** Average accuracy vs. different numbers of labeled data

### *5.3.3 Combination of feature relevance and redundancy*

In this section, we will illustrate the performance of SSMRMR feature selection algorithm through a set of contrast experiments. In this first scenario, we compare the performance of mRMR, CCLS, and SSMRMR algorithms. In addition, we compare performances of SSMRMR with different number of labeled data.

Table 6 and Fig. 10 show the performance comparison among mRMR, CCLS, and SSMRMR algorithms. It can be seen in Table 6 that in terms of averaged accuracy gains, CCLS increases 2.53 percentage points and SSMRMR increases 2.7 percentage points, compared with mRMR algorithm. SSMRMR is better than CCLS with only 0.17 percentage points. However, in terms of optimized number of features, SSMRMR decreases 8 features compared with CCLS, which means redundancy elimination can help achieve a higher degree of dimensionality reduction without accuracy decrease.

Figure 10 shows three curves of classification accuracy vs. different number of selected features. We can see that both the curves of CCLS and SSMRMR (*black and blue*) outperform the curve of mRMR (*red*). But the SSMRMR's curve increases more rapidly and achieves good performance with a small number of features.

Figure 11 shows the plots of average accuracy vs. different number of labeled data. It can be concluded that the average accuracy increased with the addition of labeled data, but wouldn't continue to increase when the labeled data reach a certain amount. Moreover, it is obvious that SSMRMR algorithm outperforms other algorithms significantly.

After the optimal feature subset has been selected, the classification is done on test data set. The results are listed in Table 7. From Table 7, it is easy to find the optimal feature subset selected from develop dataset can help improve the performance in test dataset. Through CCLS and SSMRMR still outperform other algorithms, the accuracy differences between

**Table 7** Accuracy of different algorithms on test dataset. (400 labeled segments)

| Algorithms | Spec | ReliefF | LS | CS | CLS | CCLS | SSMRMR |
|---|---|---|---|---|---|---|---|
| Accuracy | 88.87 | 90.24 | 89.77 | 89.02 | 87.35 | 91.06 | **91.11** |

algorithms are relatively small. However, the average accuracy of SSMRMR is much high than other algorithm, which means there are enough alternative optimal feature subsets of different feature numbers can be used, and we needn't worry about that the selected optimal feature subset perform good only in develop dataset.

## 6 Conclusion

In this paper, we present a feature selection algorithm under the framework of mRMR algorithm. Rather than using mutual information to measure relevance and redundancy, a new score function named CCLS was developed to evaluate the relevance of features and the GMM-based Bhattacharyya distance was used to measure the redundancy between features. The CCLS algorithm evaluated feature relevance by making full use of locality preserving ability and constraint preserving power. The GMM-based Bhattacharyya distance evaluated redundancy more appropriately and is easier to extract than MI. SSMRMR optimized the minimum redundancy condition and the maximum relevance condition simultaneously and obtain better performance not only in classification accuracy but also in dimensionality reduction.

However, there are still some limitations in our experiments. The audio data used in our experiments is collected from telephone channel where the audio types are simple and without the mixed segments (speech over music). Audio classification and segmentation under broadcast channel are more challenging for the complex audio types, lower signal-to-noise ratio, and lots of mixed segments. Thus, How to extend our work to broadcast channel will be the focus in the further work.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China (No. 61673395, No. 61403415, No. 61302107, and No. 61403224).

## References

1. Bartsch MA, Wakefield GH (2005) Audio thumbnailing of popular music using chroma-based representations. IEEE Trans Multimedia 7(1):96–104
2. Benabdeslem K, Hindawi M (2014) Efficient semi-supervised feature selection: constraint, relevance and redundancy. IEEE Trans Knowl Data Eng 26(5):1131–1143
3. Bhalerao A, Rajpoot N (2003) Selecting discriminant subbands for texture classification. in Proc. BMVC, Norwich, September 2003
4. Breiman L, Friedman JH, Olshen RA, Charles J (1984) Classification and regression trees. Wadsworth & Brooks, Pacific Grove
5. Chao YH, Wang HM, Chang RC (2005) GMM-based Bhattacharyya kernel fisher discriminant analysis for speaker recognition. in Proc. ICASSP, p 649–652
6. Choi E, Lee C (2003) Feature extraction based on the Bhattacharyya distance. Pattern Recogn 36(8):1703–1709
7. Chung FRK (1997) Spectral Graph Theory. AMS
8. Ding C, Peng HC (2003) Minimum redundancy feature selection from microarray gene expression data. in Proc. IEEE CSB, p 523–528
9. Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. J Mach Learn Res 5:845–889
10. Dy JG, Brodley CE, Kak AC, Broderick LS, Aisen AM (2003) Unsupervised feature selection applied to content-based retrieval of lung images. IEEE Trans Pattern Anal Mach Intell 25:373–378

11. Efron B, Hastie T, Johnstone I, Tibshirani R (2004a) Least angle regression. Ann Stat 25:407–449
12. Efron B, Hastie T, Johnstone I, Tibshirani R (2004b) Least angle regression. Annals of Stastics 32:407–449
13. Fukunaga K (1990) Introduction to statistical pattern recognition, 2nd edn. Academic Press, San Diego
14. Geiger JT, Schuller B, Rigoll G (2013) Large-scale audio feature extraction and SVM for acoustic scene classification. in Proc. Applications of Signal Processing to Audio and Acoustics, New Paltz, p 1–4
15. Giannakopoulos T, Pikrakis A (2014) Introduction to Audio Analysis: A MATLAB Approach, Elsevier Academic Press
16. Giannakopoulos T, Pikrakis A, Theodoridis S (2008) Gunshot detection in audio streams from movies by means of dynamic programming and Bayesian networks. in Proc. ICASSP
17. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
18. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. in proc. NIPS, Vancouver
19. Janett WW, Li Y (2009) Estimation of mutual information: A survey. in Proc. Rough Sets and Knowledge Technology, 2009, Gold Coast, Australia, July, 2009
20. Kailath T (1967) The divergence and Bhattacharyya distance measuresin signal selection. IEEE Trans Commun Technol 15(1):52–60
21. Misra H, Ikbal S, Bourlard H, Hermansky H (2004) Spectral entropy based feature for robust ASR. in Proc. ICASSP
22. Panagiotakis C, Tziritas G (2005) A speech/music discriminator based on rms and zero-crossings. IEEE Trans. on Multimedia 7(1):155–166
23. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8): 1226–1238
24. Quinlan JR (1993) C4.5:programs for machine learning. Moran Kaufmann Publishers Inc, San Francisco
25. Ramalingam T, Dhanalakshmi P (2014) Speech/music classification using wavelet based feature extraction Techiques. J Comput Sci 10(1):34–44
26. Reyes-Aldasoro CC, Bhalerao A (2006) The Bhattacharyya space for feature selection and its application to texture segmentation. Pattern Recogn 39(5):812–826
27. Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn 53:23–69
28. Ross BC (2014, Feb.) Mutual information between discrete and continuous data sets. PLoS ONE [Online] 9(2):e87357 Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0087357
29. Scheirer E, Slaney M (1997) Construction and evaluation of a robust multifeature speech/music discriminator. in Proc. ICASSP
30. Song L, Smola A, Gretton A, Bedo J, Borgwardt K (2012) Feature selection via dependence maximization. J Mach Learn Res 13:1393–1434
31. Suzuki T, Sugiyama M, Tanaka T (2009) Mutual information approximation via maximum likelihood estimation of density ratio. in Proc. IEEE International Symposium on Information Theory
32. Wang RY (2011) Research on audio classification under complex environment. Ph. D. dissertation, Beijing University of Posts and Telecommunications
33. Xu Z, Jin R, Lyu MRR, King K (2009) Discriminative semi-supervised feature selection via manifold regularization. Proc. 21st Int'l Joint Conf. Artificial Intelligence (IJCAI)
34. Yang XK, He L, Qu D, Zhang WQ, Johnson MT (2016) Semi-supervised feature selection for audio classification based on constraint compensated Laplacian score. EURASIP Journal on Audio, Speech, and Music Processing. doi:10.1186/s13636-016-0086-9
35. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn 5: 1205–1224
36. Zhang D, Chen S, Zhou Z (2008) Constraint score: a new filter method for feature selection with pairwise constraints. Pattern Recogn 41(5):1440–1451
37. Zhao Z, Liu H (2007a) Semi-supervised feature selection via spectral analysis. in proc. SIAM Int. conf. Data Mining, Tempe, p 641–646
38. Zhao Z, Liu H (2007b) Spectral Feature Selection for Supervised and Unsupervised Learning. in Proc. 24th international conference on Machine learning (ICML), p 1151–1157
39. Zhao Z, Liu H (2012) Spectral feature selection for data mining (data mining and knowledge discovery series). Chapman and Hall-CRC, Boca Raton
40. Zubair S, Yan F, Wang W (2013) Dictionary learning based sparse coefficients for audio classification with max and average pooling. Digital Signal Process 23(5):960–970

**Xu-kui Yang** was born in Fujian, China, in 1988. He received the B.S. and M.S. degrees in information and communication from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2011 and 2014, respectively. He is currently working towards the Ph.D. degree on speech recognition at the Zhengzhou Information Science and Technology Institute. His research interests are in speech signal processing, continuous speech recognition, and machine learning.

**Liang He** was born in Liaoning, China, in 1981. He received the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, in 2011. He is an Assistant professor in the Department of Electronic Engineering, Tsinghua University. His research interest covers speaker recognition and language recognition.

**Dan Qu** received the M.S. degree in communication and information system from Xi'an Information Science and Technology Institute, Xi'an, China, in 2000 and the Ph.D. degree in information and communication engineering from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2005. She is an Associate Professor at the Zhengzhou Information Science and Technology Institute. Her research interests are in speech signal processing and pattern recognition.

**Wei-Qiang Zhang** was born in Hebei, China, in 1979. He received the B.S. degree in applied physics from University of Petroleum, Shandong, in 2002, the M.S. degree in communication and information systems from Beijing Institute of Technology, Beijing, in 2005, and the Ph. D degree in information and communication engineering from Tsinghua University, Beijing, in 2009. He is an Associate Professor at the Department of Electronic Engineering, Tsinghua University, Beijing. His research interests are in the area of radar signal processing, acoustic signal processing, speech signal processing, machine learning and statistical pattern recognition.