International Journal of Multimedia and Ubiquitous Engineering
Vol.12, No.4 (2017), pp.53-70
http://dx.doi.org/10.14257/ijmue.2017.12.4.06

A SVM Active Learning Algorithm Based on Class Boundary Characteristics, and Its Application in Audio Classification

Yan Leng[1*], Nai Zhou[1], Chengli Sun[2], Xinyan Xu[3], Qi Yuan[1], Yunxia Liu[4], Dengwang Li[1] and Zhiyuan Guo[5]

[1]Shandong Province Key Laboratory of Medical Physics and Image Processing Technology, Institute of Biomedical Sciences, School of Physics and Electronics, Shandong Normal University, Ji'nan, 250014, China
[2]School of Information, Nanchang Hangkong University, Nanchang, 330063, China
[3]Department of Computer Science and Technology, Shandong College of Electronic Technology, Ji'nan, 250014, China
[4]Shandong Provincial Key Laboratory of Network Based Intelligent Computing, School of Information Science and Engineering, University of Jinan, Ji'nan, 250014, China
[5]China Electronics Technology Group Corporation No.38 Research Institute
lyansdu@163.com

Abstract

Audio classification means recognizing different types of audio events in the audio documents. One difficulty of audio classification is the sample labeling problem, because manual labeling is very time-consuming, and then it is usually difficult to get enough labeled samples for training. To reduce the manual labeling workload, one effective way is to use the SVM (Support Vector Machines) active learning technology. SVM is a discriminant classifier which is only interested in class boundary samples, and then samples on class boundary are more informative to SVM. To this end, in this work we propose to select unlabeled samples based on class boundary characteristics, and propose a new SVM active learning algorithm. We summarize 3 characteristics of class boundary, i.e. 1) the class boundary lies in a low-density region; 2) the class boundary region is confusing; 3) there exists redundancy in the class boundary region. We use the proposed active learning algorithm to resolve the sample labeling problem of audio classification. Experimental results on two real datasets verify that the samples selected based on the above class boundary characteristics are very informative, and compared with another two SVM active learning algorithms, the proposed algorithm can obtain the best performance with the least manual labeling workload, so, our proposed algorithm can effectively reduce the manual labeling workload, and then it can resolve the sample labeling problem effectively.

Keywords*: Active learning; Support vector machines; Cluster assumption; Redundancy; Audio classification*

1. Introduction

In the field of multimedia classification, recently, audio classification (AC) has been attracting more and more attention [1-2]. Compared with other media types, such as image, audio has many unique advantages. For example, audio is not affected by illumination condition, and it is not sensitive to occlusion. AC is one kind of multimedia

[1] *Corresponding Author

ISSN: 1975-0080 IJMUE

classification; it classifies the multimedia by using its audio information, specifically, AC means to recognize the audio events contained in an audio document, and then assign them the event labels. One barrier that restricts the development of AC is the sample labeling problem. The reason is that manual labeling is very time-consuming, and then when training the classification model, there are usually not enough labeled samples.

Many technologies have been proposed to resolve the sample labeling problem, among them, active learning (AL) [3] and semi-supervised learning (SSL) [4] are the two most popular ones. Both AL and SSL are an iterative process. In each iteration, AL selects the most informative samples, and then asks the expert for their labels; while for SSL, it selects the sample that has the highest confidence, and assigns it the label predicted by the machine itself. In this work we adopt AL to resolve the sample labeling problem, while use the cluster-assumption of SSL to help to find the informative samples. Cluster-assumption points out that samples in the same cluster are very likely to have the same class label, which can reflect one aspect of class boundary characteristics, beside that, we also summarize another two class boundary characteristics. Based on these characteristics, we design a method to find out the informative samples, and propose a new SVM (Support Vector Machines) active learning algorithm.

Experimental results on two real datasets show that compared with another two SVM active learning algorithms, the proposed algorithm can achieve the best classification results with the least manual labeling workload, thus in practical applications, it can effectively reduce manual labeling workload. Besides, the propose AL algorithm is not limited to audio classification, it can be extended to other classification fields, such as image classification and text classification *etc.*

The rest of this paper is organized as follows. Section 2 briefly introduces the theoretical foundations related to this work; Section 3 discusses the related work; Section 4 presents the proposed SVM active learning algorithm; Section 5 shows the experimental results and analysis, and Section 6 draws the conclusions.

2. Theoretical Foundations

2.1. Semi-supervised Learning

SSL is an iterative learning process. In each iteration, it selects the sample which is most confident by the current classifier, and adds the class label by the machine itself. Therefore, SSL does not need any human involvement. There are two basic assumptions in SSL: cluster-assumption and manifold assumption. In this work we only focus on the cluster-assumption. Cluster-assumption points out: samples in the same cluster are very likely to have the same class label. Cluster-assumption is equivalent to the low density separation assumption, and then according to the low density separation assumption, cluster-assumption can be re-expressed as: the class boundary should pass through the low-density region.

2.2. Active Learning

AL is also an iterative learning process. In order to reduce manual labeling workload, in each iteration, AL only selects the informative samples, and asks the expert for labels. Among so many AL technologies, the pool-based AL is the most popular one. Pool-based AL assumes that a small labeled training set and a large pool of unlabeled samples can be obtained, and then in each iteration, it selects the informative unlabeled samples from the pool, and asks the expert for labels; after that, the newly labeled samples are put into the labeled training set, and then the updated labeled training set is used to retrain the classifier. The workflow of the pool-based AL is expressed in Figure1

The key point of pool-based AL is step2, *i.e.* how to define informative samples and how to find them. Some research adopts the sequential AL method to find out the

informative samples. Sequential AL is one kind of AL which selects a single most informative sample from U in each iteration. For sequential AL, since in each iteration only one sample is selected and labeled, it is very time consuming. Opposite to sequential AL, there is another kind of AL called batch-mode AL. Batch-mode AL is one kind of AL which selects several informative samples in each iteration. Compared with sequential AL, for batch-mode AL, selecting several samples at a time can save training time, but on the other hand it would cause redundancy, and redundancy would increase computational complexity without corresponding benefit.

Input *the initial labeled training set L*
the unlabeled training set U
Output *the SVM classifier*

Step 1. *With the initial labeled training set L, an initial classifier is trained to seed the AL process*
Step 2. *The informative samples are selected from U*
Step 3. *The informative samples are hand-labeled by the expert and are put into L*
Step 4. *The classifier is retrained by the updated training set L*
Step 5. *Repeat step2~4 until the stopping criterion is satisfied*
Step 6. *Output the SVM classifier*

Figure 1. The Workflow of Pool-based Active Learning

2.3. SVM

SVM (support vector machines) is a discriminant classification model which has been widely used in audio classification. The decision function of SVM is:

$$f(x) = w^T x + b \tag{1}$$

Where w is the weight vector and b is the bias. $f(x)=0$ denotes the separating hyperplane. The area between the hyperplane $f(x)=-1$ and the hyperplane $f(x)=+1$ is called margin band. The goal of SVM is to find a hyperplane which can separate the training data with the maximal margin. Assuming there is a training set containing N labeled samples (x_i, y_i), $i=1,\ldots,N$, $x_i \in R^d$ is the training sample, and $y_i \in \{+1,-1\}$ is the label, then the goal of SVM can be expressed as follows:

$$\min_{w,\xi_i} \ 1/2\, w^T w + C\sum_{i=1}^{n} \xi_i \tag{2}$$

$$\text{Subject to } y_i\left((w \cdot x_i) + b\right) \geq 1 - \xi_i, \quad i = 1, \ldots, n \tag{3}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, n \tag{4}$$

Where ξ_i is the slack variable and C is a positive penalty coefficient. According to the Lagrange optimization theory, the above optimization problem can be converted into the following dual representation:

$$max \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K\left(x_i, x_j\right) \tag{5}$$

$$\text{Subject to } \sum_{i=1}^{N} y_i \alpha_i = 0 \quad i = 1, \ldots, N \tag{6}$$

$$0 \leq \alpha_i \leq C \tag{7}$$

Where α_i denotes the Lagrange multiplier. $K(\cdot,\cdot)$ is a kernel function which satisfies the Mercer conditions. The role of $K(\cdot,\cdot)$ is to project samples into a high-dimensional feature

space where the samples are linearly separable. According to the solution of the above optimization, the decision function of SVM can be re-expressed as:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{8}$$

Most of the α_is would be zero; for the non-zero α_i , the sample x_i corresponding to it is called support vector (SV). For a test sample x, its class label is predicted as $sgn(f(x))$.

3. Related Work

SVM active learning is an effective way to reduce manual labeling workload, and recently it has been widely studied. According to different criteria, SVM active learning can be divided into different types, including binary-class SVM active learning, multi-class SVM active learning [5-6], single-label SVM active learning [5], multi-label SVM active learning [7], sequential SVM active learning and batch-mode SVM active learning [8-10] *etc.*

Binary-class SVM active learning is the AL whose basic classification model is the binary-class SVM; likewise, multi-class SVM active learning is the AL whose basic classification model is the multi-class SVM. Much research has been done on binary-class SVM active learning, the study on multi-class SVM active learning is relatively little. Research in [5] and [6] are two examples of multi-class SVM active learning. In [5], the authors combined spectral and spatial information of images to select samples for manual labeling. In [6], several unlabeled samples are first selected by AL, after that, the model mines the pattern classes for unlabeled samples by computing the difference between the labeled and unlabeled samples; besides, AL is also used to select compatible, rejected and uncertain samples. Through the above strategies, the research in [6] can translate an unlabeled multi-classification problem into a supervised multi-classification problem.

Single-label SVM active learning is the AL which assigns a single label to one sample; likewise, multi-label SVM active learning is the AL which assigns multiple labels to one sample. In real life, there is often the case that for one sample, several labels are needed to describe it. Taking audio classification for example, for an audio clip, there may be several audio events happening simultaneously, and then multiple labels should be annotated for it. Much effort has been done on single-label SVM active learning, while research on multi-label SVM active learning is still little. The work in [7] belongs to multi-label SVM active learning. In [7], two multi-label AL strategies were proposed, *i.e.* max-margin prediction uncertainty strategy and label cardinality inconsistency strategy; these two strategies were then integrated into an adaptive framework of multi-label SVM active learning.

Sequential SVM active learning is the AL which selects a single most informative sample in each iteration, and batch-mode SVM active learning is the AL which selects several informative samples in each iteration. Nowadays, batch-mode AL is the mainstream, and sequential AL is seldom used since it is time-consuming. The work in [8-10] belongs to the batch-mode SVM active learning. In [9], Tuia *et al.* proposed to select samples according to the closeness degree between samples and the current separating hyperplane, besides, in order to reduce redundancy, the authors added the constraint that no selected sample should share the closest support vector. In this work, we denote this algorithm as SVM_{Tuia}. In [10], Patra *et al.* proposed to introduce the cluster-assumption of SSL into sample selection. They used the entropy-based histogram thresholding method to find out the low-density region, and then selected informative samples from it. This algorithm is denoted as SVM_{Patra} in this work.

In this work, our proposed new SVM active learning algorithm, denoted as SVM_{Leng}, belongs to the batch-mode AL. SVM_{Leng} has borrowed the idea of SVM_{Tuia} to reduce redundancy, and has borrowed the idea of SVM_{Patra} to find out the low-density region, but different from SVM_{Tuia}, when reducing redundancy, beside the constraint proposed in

SVM_{Tuia}, SVM_{Leng} has added another constraint, and different from SVM_{Patra}, SVM_{Leng} has expanded the sample selection region to include not only the low-density region proposed in SVM_{Patra}, but also the confusing regions.

4. SVM Active Learning Algorithm Based on Class Boundary Characteristics

4.1. The Class Boundary Characteristics

From the principle of SVM it can be seen that SVM is only interested in samples on the class boundary, so, for SVM, samples on the class boundary are more informative. In this work, we propose to select such informative samples based on class boundary characteristics. In order to show these characteristics intuitively, here we take two classes of samples both of which obey the Gaussian distribution as an example to discuss these characteristics (Figure2). We summarize the following three characteristics of class boundary, just as that shown in Figure2:

(1) The class boundary lies in a low-density region

Samples of the same class tend to be more close to each other, and then the class boundary region turns out to be a region with low-density.

(2) The class boundary region is confusing

In the class boundary region, samples of different classes tend to be mixed with each other, and then samples on the class boundary are easy to be misclassified.

(3) There exists redundancy in the class boundary region

In the class boundary region, some samples are very close to each other (such as sample A and sample B in Figure2), and then it would cause redundancy.

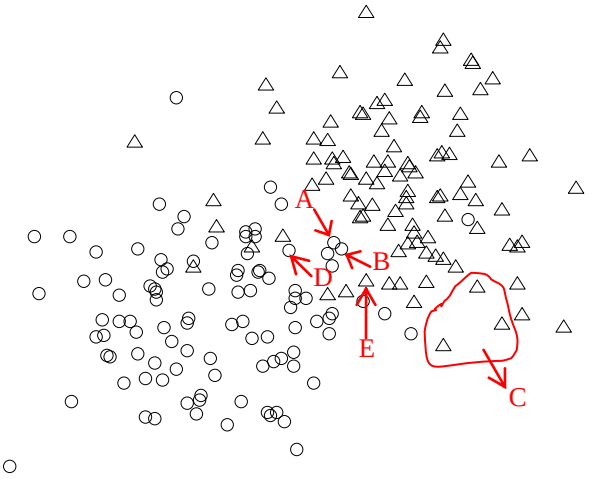


Figure 2. The Sample Distribution of Two Classes Both of Which Obey the Gaussian Distribution

4.2. Method and Principle

In this section, based on the class boundary characteristics discussed above, we design the SVM active learning algorithm as follows.

(1) The Class Boundary Lies in a Low-density Region

The cluster-assumption of SSL points out that the separating hyperplane should pass through a low-density region, and this is confirmed by the sample distribution in Figure2, so using the cluster-assumption of SSL to guide the sample selection of AL is reasonable. In this work, we use the cluster-assumption in this way: during the AL process, in some iterations, the separating hyperplane may deviate from the class boundary seriously, in this case, we will adjust it towards the class boundary through selecting the unlabeled samples from the low-density region.

In order to find out the low-density region, here we adopt the method proposed by Patra *et al*. [10]. In each iteration of AL, all samples, including the labeled ones and unlabeled ones, are first classified by the SVM classifier; then the margin band of SVM is equally divided into *M* bins; for each bin, the ratio of the samples in it to the samples in the margin band is calculated, in this way, a histogram can be constructed for the margin band; with this histogram, an entropy-based histogram-thresholding technique is used to find out the bin on which the maximum entropy is achieved, and this bin is then taken as the low-density region. The principle of entropy-based histogram-thresholding technique is as follows:

Let Ω1, Ω2 denote the two classes, *H*: the histogram constructed for the margin band, p_i: the histogram value of the *i*-th bin, *t*: the *t*-th bin of the margin band which takes values from {1,…, *M*}, then the entropy of a class can be expressed as a function of *t*:

$$E_{\Omega 1}(t) = -\sum_{i=1}^{t} \frac{p_i}{P_{\Omega 1}(t)} \log\left(\frac{p_i}{P_{\Omega 1}(t)}\right) \tag{9}$$

$$E_{\Omega 2}(t) = -\sum_{i=t+1}^{M} \frac{p_i}{P_{\Omega 2}(t)} \log\left(\frac{p_i}{P_{\Omega 2}(t)}\right) \tag{10}$$

Where $P_{\Omega 1}(t)=\sum_{i=1}^{t} p_i$ and $P_{\Omega 2}(t)= 1-P_{\Omega 1}(t)$. The optimal threshold t_0 is defined as the *t* which can maximize $E_{\Omega 1}(t)+ E_{\Omega 2}(t)$, as that shown in formula (11). The t_0-th bin of the margin band is then taken as a low-density region.

$$t_0 = \arg\max_{t} \left\{E_{\Omega 1}(t) + E_{\Omega 2}(t)\right\} \tag{11}$$

(2) The Class Boundary Region is Confusing

The class boundary has many characteristics, and then selecting informative samples only based on its low-density characteristic is not enough. Taking region C in Figure2 for example, C is a low-density region, but when performing classification using SVM, sometimes the samples in C would not fall into the margin band, and then can not be selected. But it can be seen that samples in C locate on the class boundary, and they are very informative for training.

In addition to the low-density characteristic, the class boundary also has the confusing characteristic, because the sample distribution of two classes always overlaps on the boundary, and then samples on the boundary are very easy to be misclassified. Such misclassified samples are very informative since they are just the samples that the classifier is uncertain about. If samples in C can not be selected through the low-density characteristic, here we try to select them through the confusing characteristic. To this end, here we design to expand the sample selection region to include not only the low-density region, but also the confusing regions. To do so, first, the confusion rate η is calculated for all bins; we use $bin(t_0+m)$ to denote the (t_0+m)-th bin, *m* takes values from {1- t_0, 2-t_0,

…, -1, 1, 2, …, M- t_0}, then the confusion rate of the (t_0+m)-th bin is calculated as follows:

$$\eta(t_0+m)=\frac{\#\left(x_i \,|\, x_i \in bin(t_0+m) \bigcap y(x_i) \neq \mathrm{sgn}(f(x_i))\right)}{\#\left(x_i \,|\, x_i \in bin(t_0+m)\right)} \tag{12}$$

Here, $y(x_i)$ and sgn($f(x_i)$) denote the true class label and the predicted class label of sample x_i respectively. Since the true class labels of the unlabeled samples cannot be known in advance, here we only use the labeled samples to estimate the confusion rate.

Let R denote the expanded sample selection region, it is initialized as R=$bin(t_0)$. For a bin, if its confusion rate is larger than zero, then it is incorporated into R, otherwise it is just neglected:

$$R \Leftarrow \begin{cases} R \bigcup bin(t_0+m) & \eta(t_0+m)>0 \\ R & \eta(t_0+m)=0 \end{cases} \tag{13}$$

After the above expansion operation, R contains both the low-density region and the confusing regions, and then it can well cover the class boundary region.

(3) There Exists Redundancy in the Class Boundary Region

From Figure2 it can be seen that some informative samples in R are very close to each other, which means that there exists redundancy in R. Redundancy could increase computational complexity without corresponding benefit. In order to reduce the redundancy in R, here we restrict that the selected samples should not share the closest support vector, and at the mean time should not be the nearest neighbor of each other.

In order to reduce redundancy, Tuia *et al.* [9] proposed a restriction that the selected samples should not share the closest support vector; in this work, we think that only using this restriction is not enough. Taking sample A and sample B in Figure2 for example, assuming the closest support vector of sample A is sample D, and the closest support vector of sample B is sample E, then A and B do not share the closest support vector, but apparently these two samples are very close to each other, choosing both of them is not necessary. Therefore, in order to further reduce redundancy, beside the restriction that the selected samples should not share the closest support vector, we also restrict that the selected samples should not be the nearest neighbor of each other. Based on the above discussions, the unlabeled samples in R are selected as follows.

Let d_{ave} denote the average decision value of samples in the t_0-th bin, first, the unlabeled samples in R are sorted in ascending order according to the absolute difference between their decision values and d_{ave}. The sorted samples are denoted as {x_1, x_2, … , $x_{|R|}$}. x_1 is the sample whose decision value is closest to d_{ave}, and $x_{|R|}$ is the sample whose decision value is farthest away from d_{ave}. Then a vector δ and a scalar γ are assigned to each unlabeled sample x_i in R:

$$\delta(x_i)=\left(\delta_1(x_i),\delta_2(x_i),\cdots,\delta_{|R|}(x_i)\right) \tag{14}$$

where

$$\delta_j(x_i)=\begin{cases} 1 & if\ \ j=i\ or\ x_j=N(x_i) \\ 0 & others \end{cases} \tag{15}$$

$$\gamma(x_i)=\arg\min_{q_k \in SV} Dis(x_i,q_k) \tag{16}$$

$N(x_i)$ denotes the nearest neighbor of sample x_i in R; $Dis(\cdot,\cdot)$ denotes the Euclid distance of two samples; SV denotes the support vector set; $\gamma(x_i)$ denotes the index of support vector closest to sample x_i. For vector $\delta(x_i)$, only two of its elements are non-zero, one is $\delta_i(x_i)$ which indicates the sample x_i itself, and the other is the element which indicates the nearest neighbor of x_i, in this way, if two samples have the same δ value, then they are the nearest neighbor of each other.

During sample selection process, the sample whose decision value is closest to d_{ave} (*i.e.* x_1) is first selected, then for each sample in $\{x_1, x_2, \ldots, x_{|R|}\}$, from left to right, if the sample satisfies the following restrictions, then it is selected, otherwise it is neglected.

$$\delta(x_i) \neq \delta(x_j) \quad j = 1,...,i-1 \tag{17}$$

$$\gamma(x_i) \neq \gamma(x_j) \quad j = 1,\cdots,i-1 \tag{18}$$

The reason for selecting samples in $\{x_1, x_2, \ldots, x_{|R|}\}$ from left to right is that such operation can give priority to the samples in the low-density region. After the above operations, the selected samples are not the nearest neighbor of each other, also they do not share the closest support vector, therefore, the redundancy in R can be reduced effectively, and then the selected samples would have certain diversity.

The whole procedure of SVM_{Leng} is expressed in Figure3.

Input *the initial labeled training set L*
the unlabeled training set U
Output *the SVM classifier*

Step 1. *Train the SVM classifier with the initial labeled training set L*
Step 2. *Find the low-density region within the margin band of SVM (4.2 (1))*
Step 3. *Expand the sample selection region to include the confusing regions (4.2 (2))*
Step 4. *Reduce the redundancy that exists in the sample selection region (4.2 (3))*
Step 5. *Hand-label the selected unlabeled samples, and put them into L*
Step 6. *Retrain the classifier using the updated training set L*
Step 7. *Repeat step2~6 until satisfying the stopping criterion*
Step 8. *Output the SVM classifier*

Figure 3. The Framework of the Proposed SVM_{Leng}

5. Experimental Results and Analysis

5.1. Experimental Setting

In this section, in order to test the effectiveness of SVM_{Leng}, we use it to classify the speech in the audio documents. The algorithm is tested on two real datasets: the "Friends" dataset and the "Daily life" dataset. In the "Friends" dataset, there are 10 episodes of melodrama "Friends" [11] which contain audio events of speech, music and laugh *etc.* In the "Daily life" dataset [12], there is an audio document of 2.75-hour-long which contains audio events of speech, keystrokes and footsteps *etc.* In the annotation stage, the audio events are annotated as speech or non-speech. Each annotated segment is then segmented into 1s long clips with 50% overlap which are taken as the audio samples. During feature extraction process, the audio clips are further segmented into frames, with frame length\ shift to be 30 \10ms. For each frame, a set of features are extracted, including zero crossing rate and MFCC *etc.*, then for each clip, the mean and standard deviation of the features are computed over all the frames in it, besides, some long-time features are also extracted. The features and their dimensions are illustrated in Table1.

For both datasets, 80% of the total samples are selected randomly to serve as the training samples, and the left 20% serve as test samples. From the training set, 10% of the samples are selected randomly to serve as the initial labeled training samples, and the left 90% act as unlabeled samples. For SVM, RBF (radial basis function) is chosen as the kernel function. The parameters of SVM are determined through the grid search method.

Table 1. The Features and their Dimensions

Features	Dimension
short-time energy	1
zero crossing rate	1
Mel Frequency Cepstral Coefficients	8
sub-band energy ratio	8
sub-band spectral flux	8
brightness	1
bandwidth	1
Line Spectrum Pair	10
high zero crossing rate	1
low energy rate	1
spectrum flux	1

5.2. Classification Performance and Manual Labeling Workload

The proposed SVM_{Leng} is a batch-mode AL algorithm, in order to test its effectiveness, here we compare it with another two batch-mode AL algorithms: SVM_{Tuia} and SVM_{Patra}. Besides, we use all training samples to train a SVM classifier, denoted as fullSVM, and take its performance as a reference of the best performance that the classifier can achieve. Here we will also compare SVM_{Leng} with fullSVM. For each algorithm, the experiment is done 5 times, and the average is taken as the final result. The *F*1-score defined as follows along with the manual labeling workload are taken as the evaluation criterion.

$$F1 = \frac{2 \times recall \times precision}{\left(recall + precision \right)} \times 100\% \tag{19}$$

Where

$$precision = \frac{the\ number\ of\ \ correctly\ classified\ \ speech\ samples}{the\ number\ of\ \ samples\ that\ are\ recognized\ \ as\ \ speech} \tag{20}$$

$$recall = \frac{the\ number\ of\ \ correctly\ classified\ \ speech\ samples}{the\ total\ number\ of\ \ speech\ samples\ in\ the\ test\ \ set} \tag{21}$$

The *F*1-scores corresponding to queries in discriminating speech from non-speech on "Friends" and "Daily life" dataset are shown in Figure4 and Figure5 respectively. From Figure4 and Figure5 it can be seen that on both datasets, among the three SVM active learning algorithms: SVM_{Leng}, SVM_{Tuia} and SVM_{Patra}, it is the proposed SVM_{Leng} that performs best. On "Friends" dataset, after about 25 queries, the performance of SVM_{Leng} has exceeded that of fullSVM, while for SVM_{Tuia} and SVM_{Patra}, they need about 100 queries to achieve that. On "Daily life" dataset, SVM_{Leng} needs about 70 queries to achieve the performance of fullSVM, while SVM_{Tuia} and SVM_{Patra} need about 90 queries to do that.

Apparently, under the same classification performance, compared with SVM_{Tuia} and SVM_{Patra}, SVM_{Leng} will need less queried samples, and then it can reduce manual labeling workload; conversely, if the manual labeling workload is fixed, then under the same manual labeling workload, compared with SVM_{Tuia} and SVM_{Patra}, SVM_{Leng} will obtain better classification results. In summary, the proposed SVM_{Leng} is an effective SVM active learning algorithm, it can reduce the manual labeling workload effectively.

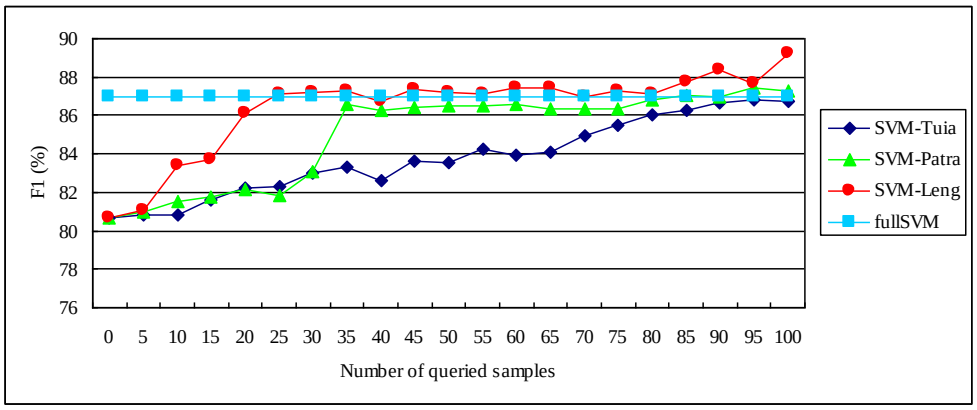


Figure 4. The *F*1-scores Corresponding to Queries when Discriminating Speech from Non-speech on "Friends" Dataset

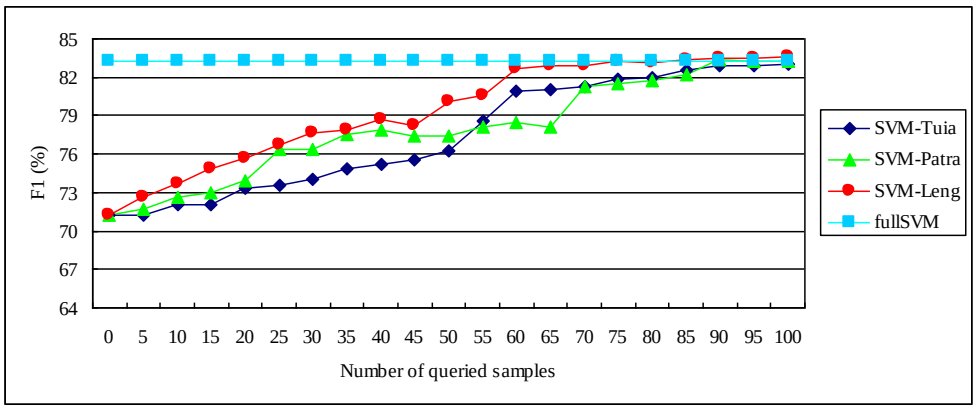


Figure 5. The *F*1-scores Corresponding to Queries when Discriminating Speech from Non-speech on "Daily life" Dataset

SVM_{Tuia} selects samples according to the closeness degree between the sample and the current separating hyperplane. The limitation of SVM_{Tuia} is that: if the current hyperplane is close to the true hyperplane, then selecting samples close to the current hyperplane would obtain good results, but if the current hyperplane is far away from the true one, then samples close to the current hyperplane may not be a good choice. Different from SVM_{Tuia}, SVM_{Leng} selects samples based on class boundary characteristics. Classification results in Figure4 and Figure5 have verified the correctness of our idea. SVM is a classifier which tries to find an optimal classification hyperplane on the class boundary, and then it is only interested in class boundary samples. Based on class boundary characteristics, SVM_{Leng} can successfully select the informative class boundary samples, and then it can reduce the manual labeling workload effectively.

SVM_{Patra} selects samples in the low-density region; it does not consider the redundancy problem. Different from SVM_{Patra}, SVM_{Leng} selects samples not only in the low-density region, but also in the confusing regions, and it has taken enough measures to reduce the redundancy. Classification results in Figure4 and Figure5 have confirmed the superiority of SVM_{Leng} over SVM_{Patra}. Compared with SVM_{Patra}, through including the confusing regions into the sample selection region, SVM_{Leng} can better cover the class boundary region, and then its selected samples are more informative. Besides, compared with SVM_{Patra}, SVM_{Leng} has reduced the redundancy through restricting that the selected samples should not share the closest support vector, and in the mean time should not be the nearest neighbor of each other. In this way, SVM_{Leng} can reduce the redundancy effectively, and then it can reduce manual labeling workload.

In this section, in order to evaluate SVM_{Leng} more comprehensively, we also compare its performance with that of SVM_{Tuia} and SVM_{Patra} after they satisfy the stopping criterion. We take the convergence of the classifier as the stopping criterion which is defined as follows.

Let
$$\eta = \frac{|F1_i - F1_{i-1}|}{F1_{i-1}} \times 100\% \tag{22}$$

$F1_i$ denotes the $F1$-score of the i-th iteration. If in a certain iteration, the η value is less than 5‰, and this case lasts for more than 5 iterations, then the classifier is defined to be convergent. For each algorithm, the experiment is done 5 times. The performance is evaluated in terms of the average and standard deviation of $F1$-score as well as manual labeling workload. The performance of the three AL algorithms on "Friends" and "Daily life" dataset is shown in Table2 and Table3 respectively.

Table 2. The Classification Performance and the Manual Labeling Workload of the Algorithms on "Friends" Dataset

Algorithms	Average $F1$(%)	Standard Deviation	Manual Labeling Workload
SVM_{Tuia}	88.96	1.05	203
SVM_{Patra}	90.11	0.65	181
SVM_{Leng}	92.08	0.32	166

Table 3. The Classification Performance and the Manual Labeling Workload of the Algorithms on "Daily life" Dataset

Algorithms	Average $F1$(%)	Standard Deviation	Manual Labeling Workload
SVM_{Tuia}	85.00	0.96	185
SVM_{Patra}	85.68	0.89	142
SVM_{Leng}	85.92	0.53	137

From Table2 and Table3 it can be seen that after satisfying the stopping criterion, among the three AL algorithms, the proposed SVM_{Leng} can achieve the best performance with the least manual labeling workload. So, for SVM_{Leng}, it needs much less manual labeling workload to achieve better performance. This illustrates that the unlabeled samples selected by SVM_{Leng} are more informative, and illustrates that our idea of selecting unlabeled samples based on class boundary characteristics is correct.

5.3. The Closeness Degree between the Separating Hyperplane Obtained by the AL Algorithm and the True Separating Hyperplane

From the principle of SVM it can be seen that the separating hyperplane is completely determined by the support vectors, and then the closeness degree between the obtained SV set and the true SV set can reflect the closeness degree between the obtained separating hyperplane and the true separating hyperplane. We use the SV set of fullSVM as an approximation of the true SV set, denoted as $V1$, and use $V2$ to denote the SV set obtained by the AL algorithm after the classifier has been convergent. Mandal *et al*. have ever proposed a distance $\mathcal{D}$ for measuring the closeness degree between $V1$ and $V2$:

$$\mathcal{D} = \frac{1}{n_{V2}} \sum_{x \in V2} \sigma(x, V1) + \frac{1}{n_{V1}} \sum_{z \in V1} \sigma(z, V2) + Dist(V1, V2) \tag{23}$$

where
$$\sigma(x, V1) = \min_{z \in V1} d(x, z) \tag{24}$$

$$\sigma(z,V2)=\min_{x\in V2} d(z,x) \tag{25}$$

$$Dist(V1,V2)=\max\{\max_{x\in V2}\sigma(x,V1),\max_{z\in V1}\sigma(z,V2)\} \tag{26}$$

n_{V1} and n_{V2} denote the number of samples in $V1$ and $V2$ respectively. Apparently, smaller $\mathcal{D}$ means that the separating hyperplane obtained by the AL algorithm and the true separating hyperplane are more close to each other. The measurement results of $\mathcal{D}$ on both datasets are shown in Table4.

Table 4. The Closeness Degree between the SV Set Obtained by the AL Algorithm and the True SV Set

Dataset	Algorithms	$\mathcal{D}$
Friends	SVM_{Tuia}	18.31
	SVM_{Patra}	17.78
	SVM_{Leng}	17.20
Daily life	SVM_{Tuia}	12.67
	SVM_{Patra}	9.97
	SVM_{Leng}	9.35

From Table4 it can be seen that on both datasets, among the three AL algorithms, SVM_{Leng} has obtained the minimal $\mathcal{D}$ value, which means that the separating hyperplane of SVM_{Leng} is most close to the true hyperplane, and therefore it can obtain the best classification results.

In each iteration, SVM_{Leng} selects unlabeled samples based on the class boundary characteristics, in this way it can ensure the separating hyperplane not to deviate too far from the true hyperplane. For SVM_{Tuia}, in each iteration, it tends to select samples which are more close to the current separating hyperplane; if the current separating hyperplane is close to the true one, then its selected samples would help to guide the current hyperplane toward the true one, but if the current separating hyperplane is far away from the true one, then its selected samples may guide the current hyperplane away from the true one. For SVM_{Patra}, in each iteration, it selects samples from the low-density region, but the low-density characteristic can not describe the class boundary completely, then selecting samples only based on the low-density characteristic can not guarantee that the samples selected in each iteration are always on the class boundary.

5.4. The Correctness of the Proposed Idea—"Expand the Sample Selection Region to Include the Confusing Regions"

In this work, the proposed SVM_{Leng} has introduced the idea proposed in SVM_{Patra} [10] to select unlabeled samples, *i.e*. selecting samples from the low-density region. Different from SVM_{Patra}, SVM_{Leng} has expanded the sample selection region; beside the low-density region, it also selects samples from the confusing regions. In order to verify the correctness of the proposed idea—"expand the sample selection region to include the confusing regions", in this section, we will compare SVM_{Leng} with another algorithm denoted as $SVM_{Leng(1)}$. $SVM_{Leng(1)}$ is an algorithm that is very similar to SVM_{Leng}. The only difference between SVM_{Leng} and $SVM_{Leng(1)}$ is that SVM_{Leng} selects samples from both the low-density region and the confusing regions, while $SVM_{Leng(1)}$ selects samples only from the low-density region. In the low-density region, $SVM_{Leng(1)}$ selects unlabeled samples according to the closeness degree between the sample and the t_0-th bin; it tends to select samples which are more close to the t_0-th bin. The performance of these two algorithms is evaluated by the $F1$-scores under different number of queries. Their classification performance on "Friends" and "Daily life" dataset is shown in Figure6 and Figure7 respectively.

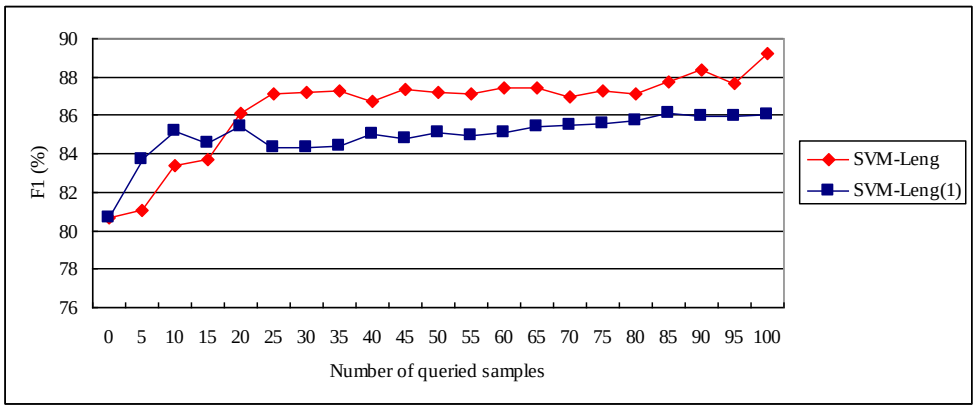


Figure 6. The Classification Performance of SVM_{Leng} and $SVM_{Leng(1)}$ on "Friends" Dataset

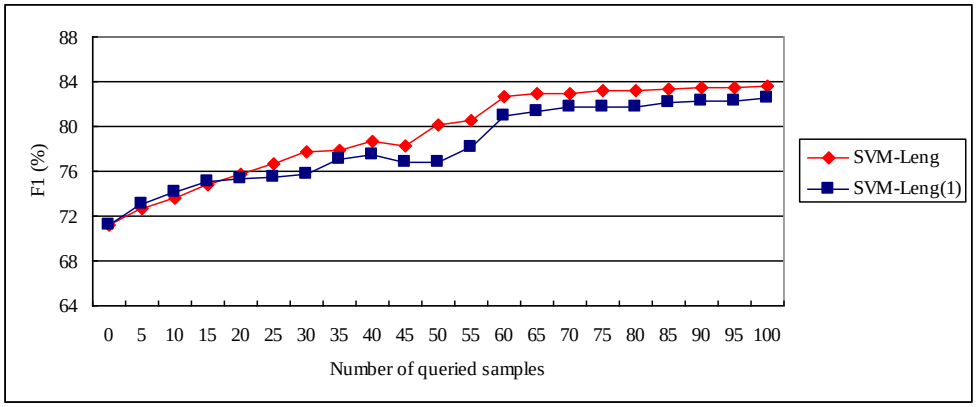


Figure 7. The Classification Performance of SVM_{Leng} and $SVM_{Leng(1)}$ on "Daily life" Dataset

From Figure6 and Figure7 it can be seen that on both datasets, on the whole, SVM_{Leng} performs better than $SVM_{Leng(1)}$, which illustrates that when selecting samples, it is very necessary to take the confusing characteristic of class boundary into consideration, and then it verifies the correctness of the proposed idea—"expand the sample selection region to include the confusing regions". Only using the low-density characteristic to describe the class boundary is not enough, the confusing characteristic is also very important; the confusing characteristic can help to find the unlabeled samples which are easy to be misclassified, such samples are very informative since they are just the samples that the current classifier is uncertain about.

On both datasets it can be seen that $SVM_{Leng(1)}$ performs a little better than SVM_{Leng} in the first few iterations, but in the later iterations, SVM_{Leng} performs much better than $SVM_{Leng(1)}$. The reason may be that: in the first few iterations, samples selected from the low-density region can adjust the current hyperplane quickly toward the true hyperplane, and then $SVM_{Leng(1)}$ can get better results, but in the later iterations, after the hyperplane has been adjusted to a much better position, samples in the confusing regions are relatively more informative, because such samples can adjust the hyperplane to fit the samples that the current classifier is still uncertain about.

5.5. The Correctness of the Proposed Idea—"The Selected Samples Should Not be the Nearest Neighbor of Each Other"

In order to further reduce redundancy, in this work we propose the idea that "the selected samples should not be the nearest neighbor of each other". In order to verify its correctness, here we will compare SVM_{Leng} with another algorithm denoted as $SVM_{Leng(2)}$. $SVM_{Leng(2)}$ is an algorithm that is very similar to SVM_{Leng}. The only difference between SVM_{Leng} and $SVM_{Leng(2)}$ is that: to reduce redundancy, SVM_{Leng} restricts that the selected samples should not share the closest support vector, and should not be the nearest neighbor of each other, while for $SVM_{Leng(2)}$, it only restricts that the selected samples should not share the closest support vector. The performance of these two algorithms is evaluated by the *F*1-scores under different number of queries. Their classification performance on "Friends" and "Daily life" dataset is shown in Figure8 and Figure9 respectively.

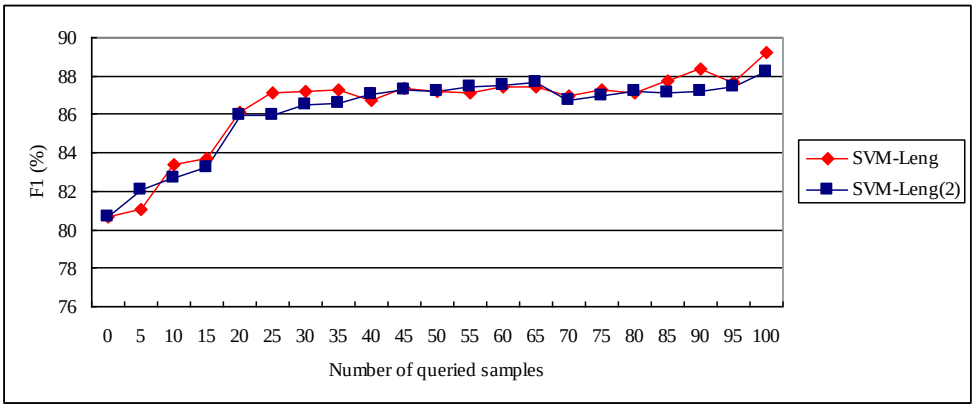


Figure 8. The Classification Performance of SVM_{Leng} and $SVM_{Leng(2)}$ on "Friends" Dataset

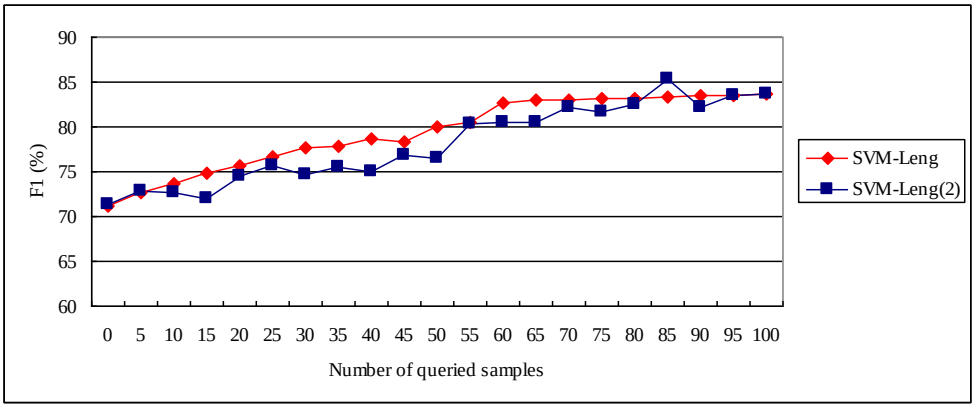


Figure 9. The Classification Performance of SVM_{Leng} and $SVM_{Leng(2)}$ on "Daily life" Dataset

From Figure8 and Figure9 it can be seen that on both datasets, on the whole, SVM_{Leng} performs better than $SVM_{Leng(2)}$, which verifies the correctness of the proposed idea that "the selected samples should not be the nearest neighbor of each other", therefore, using this idea to further reduce redundancy is very necessary. For $SVM_{Leng(2)}$, it has restricted that the selected samples should not share the closest support vector; this restriction can reduce the redundancy which is caused by the samples that are close to the same support vector, while it should be noticed that for some samples, although they do not share the

closest support vector, they could still be very close to each other, which would also cause redundancy. Such kind of redundancy can not be reduced by the restriction that "the selected samples should not share the closest support vector", but can be reduced by the restriction that "the selected samples should not be the nearest neighbor of each other".

6. Conclusions

When using SVM active learning to resolve sample labeling problem, the key point is how to define informative samples and how to find them. In this work, considering that SVM is only interested in class boundary samples, we propose that samples on class boundary are more informative, and we propose to find them based on class boundary characteristics. Experimental results have verified the effectiveness of the proposed SVM active learning algorithm, and have verified the correctness of our proposed innovative ideas, *i.e*. the sample selection region should be expanded to include the confusing regions and the selected samples should not be the nearest neighbor of each other.

Though the proposed SVM active learning algorithm has obtained satisfying results, there still exists much improvement space: 1) the proposed algorithm needs an initial labeled training set to train an initial classifier, then how to establish this initial labeled training set is a subject that needs to be carefully studied. A good initial labeled training set will make the classifier have a good initial performance, and then would give the AL algorithm a good starting point; 2) in this work we summarize 3 class boundary characteristics, in fact, some other characteristics can continue to be mined to help to find out the informative samples.

In this work, the proposed SVM active learning algorithm has been used to resolve the sample labeling problem of audio classification. In fact, it is not limited to audio classification; it can also be applied to other classification fields in which the labeled samples are difficult to get, while the unlabeled samples are easy to obtain.

Acknowledgments

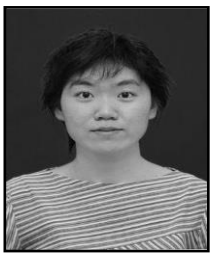
This work has been jointly supported by the Project of National Natural Science Foundation of China (No. 61401259, No. 61362031, No. 61471226, No. 61401258, No. 61501283, No. 61203239, No. 61305015), China Postdoctoral Science Foundation Funded Project (No. 2015M582128, No. 2015M582129, No. 2015M580591), Natural Science Foundation for Distinguished Young Scholars of Shandong Province (JQ201516), Natural Science Foundation of Shandong Province (ZR2013FQ019, ZR2015PF012), and The International Science and Technology Cooperation Program of China (No. 2014DFA11580).

References

[1] Y. Leng, C. Sun, X. Xu, Q. Yuan, S. Xing, H. Wan, J. Wang and D. Li, "Employing Unlabeled Data to Improve the Classification Performance of SVM, and Its Application in Audio Event Classification'', Knowledge-Based Systems, vol. 98, no. C, **(2016)**, pp. 117-129.
[2] Y. Leng, C. Sun, C. Cheng, X. Xu, S. Li, H. Wan, J. Fang and D. Li, "Classification of Overlapped Audio Events Based on AT, PLSA, and the Combination of Them", Radioengineering, vol. 24, no. 2, **(2015)**, pp. 593-603.
[3] B. Settles, "Active Learning", Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, no. 1 (**2012**), pp. 1-114.
[4] X. Zhu, "Semi-Supervised Learning Literature Survey", Computer Science, vol. 37, no. 1, **(2008)**, pp. 63-77.
[5] E. Pasolli, F. Melgani and D. Tuia, "SVM Active Learning Approach for Image Classification Using Spatial Information", IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 4, **(2014)**, pp. 2217 -2233.
[6] H. Guo and W. Wang, "An Active Learning-Based SVM Multi-Class Classification Model", Pattern Recognition, vol. 48, no. 5, **(2015)**, pp. 1577-1597.

[7] X. Li and Y. Guo, "Active Learning with Multi-Label SVM Classification", Proceedings of the 23th International Joint Conference on Artificial Intelligence, Beijing, China, **(2013)**, August 3-9.
[8] M. Goudjil, M. Koudil, N. Hammami, M. Bedda and M. Alruily, "Arabic Text Categorization Using SVM Active Learning Technique: An Overview", Proceedings of 2013 World Congress on Computer and Information Technology, Sousse, Tunisia, **(2013)**, June 22-24.
[9] D. Tuia, , F. Ratle, F. Pacifici, M.F. Kanevski and W.J. Emery, "Active Learning Methods for Remote Sensing Image Classification", IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 7, **(2009)**, pp. 2218 -2232.
[10] S. Patra and L. Bruzzone, "A Fast Cluster-assumption Based Active-learning Technique for Classification of Remote Sensing Images", IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 5, **(2011)**, pp. 1617-1626.
[11] Y. Leng, X. Xu and G. Qi, "Combining Active Learning and Semi-supervised Learning to Construct SVM Classifier", Knowledge-Based Systems, vol. 44, no. 1, **(2013)**, pp. 121-131.
[12] Y. Leng, G. Qi and X. Xu, "A BIC Based Initial Training Set Selection Algorithm for Active Learning and Its Application in Audio Detection", Radioengineering, vol. 22, no. 2, **(2013)**, pp. 638-649.

Authors

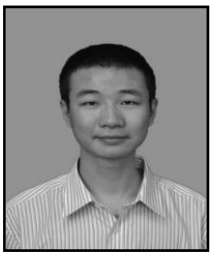


Yan Leng, She received the B.S., M.S. degrees from Shandong University (SDU), Ji'nan, China in 2003 and 2006 respectively, and received the Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 2012. Now she works as a lecturer at the School of Physics and Electronics, Shandong Normal University in Ji'nan, China. Her research interests include audio classification, audio detection and audio scene recognition.

Nai Zhou, He received the B.S. degree from Shandong Normal University (SDNU), Ji'nan, China in 2015. Now he is studying for a master's degree in SDNU. His research interests include audio classification, audio detection and machine learning.

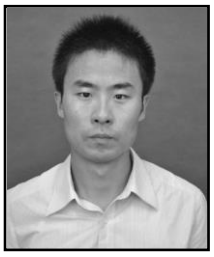
Chengli Sun, He received the B.S. degree in electronics engineering from Zhongbei University, Taiyuan, China, in 1999, an-d the Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunication (BUPT), Beijing, China in 2008. Now he works as an associate professor at the School of Information, Nanchang Hangkong University in Nanchang, China. His research interests include audio classification, speech recognition and speech enhancement.

Xinyan Xu, She received the M.S. degree from Shandong University (SDU), Jinan, China in 2006. Her research interests include audio classification and medical image processing.



Qi Yuan, He received the B.S. degree from Tianjin University of Technology and Education, Tianjin, China in 2009, and received the Ph.D. degree from Shandong University, Jinan, China in 2014. Now, he works as a lecturer at the School of Physics and Electronics, Shandong Normal University in Jinan, China. His research interests include signal processing theory and application.

Yunxia Liu, She received the B.S., M.S. and Ph.D. degrees from the school of Information Science and Engineering of Shandong University, Jinan, China in 2004, 2007 and 2012. She worked as a postdoctoral researcher in School of Control Science and Engineering of Shandong University from 2012 to 2015. Now she works as an associate professor at the School of Information Science and Engineering at University of Jinan. Her research interests include image processing, pedestrian detection and visual tracking.



Dengwang Li, He received the B.S., Ph.D. degrees from Shandong University, Jinan, China in 2006 and 2011 respectively. Now he works as an associate professor at the School of Physics and Electronics, Shandong Normal University in Jinan, China. His research interests include signal processing and medical image processing.

Zhiyuan Guo, She received the B.S. degree from Automation School of Beijing University of Posts and Telecommunications (BUPT), China in 2008, and the M.S., Ph. D. degrees from Pattern Recognition and Intelligent Systems Lab of BUPT in 2010 and 2013 respectively. Now she works as an engineer at China Electronics Technology Group Corporation No.38 Research Institute. Her research interests include speech retrieval, speech signal processing.