Multimedia Systems 8: 482–492 (2003)
Digital Object Identifier (DOI) 10.1007/s00530-002-0065-0

Multimedia Systems

Content-based audio classification and segmentation by using support vector machines

Lie Lu, Hong-Jiang Zhang, Stan Z. Li

Microsoft Research Asia 5F Beijing Sigma Center, No.49 Zhichun Road Hai Dian District, Beijing, 100080, China
(e-mail: {llu,hjzhang,szli}@microsoft.com)

Abstract. Content-based audio classification and segmentation is a basis for further audio/video analysis. In this paper, we present our work on audio segmentation and classification which employs support vector machines (SVMs). Five audio classes are considered in this paper: silence, music, background sound, pure speech, and non-pure speech which includes speech over music and speech over noise. A sound stream is segmented by classifying each sub-segment into one of these five classes. We have evaluated the performance of SVM on different audio type-pairs classification with testing unit of different-length and compared the performance of SVM, K-Nearest Neighbor (KNN), and Gaussian Mixture Model (GMM). We also evaluated the effectiveness of some new proposed features. Experiments on a database composed of about 4-hour audio data show that the proposed classifier is very efficient on audio classification and segmentation. It also shows the accuracy of the SVM-based method is much better than the method based on KNN and GMM.

Key words: Audio content analysis, audio classification and segmentation, support vector machines

1. Introduction

Audio data is an integral part of many multimedia applications. A fundamental step for further audio analysis and content understanding is to automatically classify or segment a long audio stream based on its content [1,2]. It is of critical importance in audio indexing and retrieval, and video content analysis.

Content-based classification of audio data is essentially a pattern recognition problem in which there are two basic issues: feature selection and classification base on the selected features. In terms of the former issue, an effective representation should be able to capture the most significant properties of the audio signals, robust under various circumstances and general enough to describe various audio classes. To tackle the latter issue, a good classifier is crucial.

There have been many studies on audio segmentation and classification using different features and different methods. Pfeiffer, et al. [3], presented a theoretic framework and the application of automatic audio content analysis using a set of perceptual features. Saunders [4] presented a speech/music classifier based on simple features such as zero crossing rate and short time energy for radio broadcast. When a window size of 2.4 s was used, the reported accuracy rate would be 98%. Scheirer et al. [5] introduced many more features into audio classification and performed experiments with different classification models. When using a window of the same size (2.4 s), the reported error rate would be 1.4%. However, in spite of these research efforts, high accuracy audio classification is only achieved for the simple cases such as speech/music discrimination. It is also found that methods based on such simple features cannot work well when a smaller window is used or more audio classes such as environment sounds are taken into consideration.

Many other works have been done to enhance audio classification algorithms to discriminate more classes. In the work by Zhang and Kuo [7], with a heuristic-based model, many features including pitch tracking methods are introduced to discriminate audio recordings into more classes, such as songs, speeches over music. An accuracy of above 90% is reported. Srinivasan et al. [8], try to detect and classify audio that consists of mixed classes, such as combinations of speech and music together with environment sound. The reported accuracy of classification is over 80%. In [9], an algorithm of audio classification and segmentation is presented, where speech, music, environment sound, and silence are discriminated with one-second window. An accuracy of above 96% is reported. However, in these works, rule-based classifier is used for audio classification and segmentation; these systems require threshold setting. But threshold is very difficult to set, and it should be adjusted for various circumstances. This makes the rule-based approach not general to fit different applications.

The researches trying other classifiers have also been reported. In [12], a simple nearest neighbor rule (NN) is used as its classifier. In [13], an artificial neural network (ANN) is used to classify TV programs into different categories by using perceptual features. In [5], some different classifiers, which include Gaussian Mixture Model (GMM) and K-Nearest Neighbor (KNN) are used and compared on audio classification and segmentation. In [6], audio recordings are classified into speech, silence, laughter and non-speech sounds utilizing a

Hidden Markov Model (HMM) in order to segment recordings of discussions at meetings. Some new classification algorithms are also proposed for audio classification, such as Nearest Feature Line [11] and hybrid method which combines VQ and rule-based method with multiple classifying steps [9].

Support Vector Machine (SVM) [17,18] is a very efficient classifier in pattern recognition. Compared to HMM, a generative model, SVM learns an optimal separating hyper-plan to minimize the probability of misclassification. It seems more suitable than HMM for classification. However, few works have applied support vector machines to audio classification and segmentation. The feature distribution of audio data is so complicated that different classes may have overlapping or interwoven areas. A kernel based SVM is well suited to handle such a situation. In our previous work [10], we tried SVM on audio classification and segmentation with one-second windows, and good performance was achieved. In this paper, we will further evaluate the performance of support vector machine using different classification or segmentation units and compare its performance with other popular classifiers such as k-nearest neighbor (KNN) and Gaussian Mixture Model (GMM). We will also present a comprehensive evaluation of the effectiveness of each audio feature. In this study, we classify the audio clips or segment an audio stream into five classes: silence, pure speech, non-pure speech, environment sound, and music.

In reference to the two issues mentioned above, the major contributions of the work presented in this paper are the following:

1) Propose a set of new features which could further improve the performance of the previous audio classification systems, including our own earlier works. The effectiveness of the proposed new features has been tested in the suggested SVM classification framework.
2) Propose an application framework of SVM in audio classification and segmentation.
3) Present a comprehensive evaluation and analysis of the performance of SVM on audio classification and segmentation.

The rest of this paper is organized as follows. Section 2 describes how an audio clip is represented by low level perceptual and cepstral feature. Section 3 gives an overview of linear and kernel SVM. In Sect. 4, a method for multi-class classification is discussed. In Sect. 5, an audio segmentation algorithm is presented. Finally, in Sect. 6, experiments and evaluations on a 4-hour database are given.

2. Audio feature selection

An important step of audio classification is feature selection. In order to obtain high accuracy for classification and segmentation, it is critical to select good features that can capture the temporal and spectral characteristics of audio signal. Based on the work [11], the features are divided into two types: (i) mel-frequency cepstral coefficients (MFCCs), and (ii) perceptual features. These features are combined as one feature vector after normalization.

Before feature extraction, an audio signal is converted into a general format, which is 8 KHz, 16-bit, mono-channel. Then, it is pre-emphasized with parameter 0.98 to equalize the inherent spectral tilt and then divided into non-overlapping sub-clips. A sub-clip is used as the classification unit, and segmentation is then performed based on the classification results. Classification performances with different durations of sub-clip are tested in our experiments. The sub-clip is further divided into non-overlapping 25 ms-long frames for feature extraction.

In our method, 8 order MFCCs are used as suggested by [11]. The perceptual features we selected include: zero crossing rates (ZCR), short time energy (STE), sub-band powers distribution, brightness, bandwidth, spectrum flux (SF), band periodicity (BP), and noise frame ratio (NFR). While most of these features are frequently used, some are newly introduced, such as SF, BP and NFR. The definitions of these features are given below.

2.1. Mel-frequency cepstral coefficients

These are computed from FFT. The log spectral coefficients are perceptually weighted by a non-linear map of the frequency scale, which is called Mel-scaling, using a triangular band-pass filter bank. Then, the Mel-weighed spectrum is transformed into MFCC with the COS transformation [14].

$$c_n = \sqrt{\frac{2}{K}} \sum_{k=1}^{K} (\log S_k) \cos[n(k-0.5)\pi/K] \quad n=1,2,\ldots L \tag{1}$$

where K is the number of band-pass filters, S_k is the Mel-weighted spectrum after passing kth triangular band-pass filter, and L is the order of the cepstrum. In our method, 8-order MFCCs are used, that is $L = 8$.

MFCC is commonly used in speech recognition system. Because of its good discriminating ability, it is also used in audio classification system [1,11,15].

2.2. Zero-crossing rate

Zero-Crossing Rate is defined as the number of time-domain zero-crossings within a frame. It is a simple measure of the frequency content of a signal:

$$\mathrm{ZCR} = \frac{1}{2(N-1)} \sum_{m=1}^{N-1} |\mathrm{sgn}[x(m+1)] - \mathrm{sgn}[x(m)]| \tag{2}$$

where $\mathrm{sgn}[.]$ is a sign function and $x(m)$ is the discrete audio signal, $m = 1 \ldots N$.

In general, speech signals are composed of alternating voiced sounds and unvoiced sounds in the syllable rate, while music signals do not have this kind of structure. Hence, for the speech signal, its variation of zero-crossing rate will be in general greater than that of music signals. ZCR is a good discriminator between speech and music. Considering this, many systems [4,5,7–9] have used ZCR for audio classification.

2.3. Short time energy and sub-band energy distribution

Short Time Energy (STE) is the total spectrum power of a frame. In our scheme, its logarithm is used:

$$\mathrm{STE} = \log\left(\int_0^{w_0} |F(x)|^2 dw\right) \tag{3}$$

where $F(w)$ denotes the Fast Fourier Transform (FFT) coefficients, $|F(w)|^2$ is the power at the frequency w, and w_0 is the half sampling frequency.

The frequency spectrum is divided into four sub-bands with intervals $[0, \frac{w_0}{8}]$, $[\frac{w_0}{8}, \frac{w_0}{4}]$, $[\frac{w_0}{4}, \frac{w_0}{2}]$ and $[\frac{w_0}{2}, w_0]$. The ratio between sub-band power and total power in a frame is defined as:

$$D = \frac{1}{\mathrm{STE}} \int_{L_j}^{H_j} |F(w)|^2 dw \tag{4}$$

where L_j and H_j are lower and upper bound of sub-band j respectively.

STE is an effective feature, especially for discriminating speech from music signals. In general, there are more silence frames in speech than in music; thus, the variation of STE measure will be much higher for speech than that for music. Since the frequency characteristics are very different between human voice and music apparatus, Sub-Band Energy Distribution is also a good discriminator for speech and music. These two features are also used commonly [4,5,7,8,12,13].

2.4. Brightness and bandwidth

The brightness is the frequency centroid of the spectrum in a frame, it can be defined as:

$$w_c = \frac{\int_0^{w_0} w|F(w)|^2 dw}{\int_0^{w_0} |F(w)|^2 dw}. \tag{5}$$

Bandwidth is the square root of the power-weighted average of the squared difference between the spectral components and frequency centroid:

$$B = \sqrt{\frac{\int_0^{w_0} (w - w_c)^2 |F(w)|^2 dw}{\int_0^{w_0} |F(w)|^2 dw}}. \tag{6}$$

Brightness and Bandwidth represent the frequency characteristic, and they have shown effectiveness in many audio classification systems [5,11–13].

These features are extracted from one frame, and their means and standard deviations in one audio sub-clip are computed to represent the feature of an audio sub-clip. We will use these features as a baseline system. But there still exists some misclassification. For example, some strong periodicity environment sounds are easily classified into music; noisy speech is also easily classified into music. In order to further improve the performance of baseline system, we also use a few new features, which are described in detail below.

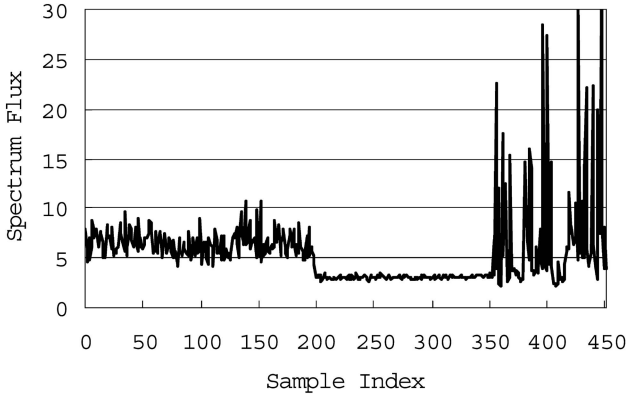

Fig. 1. The spectrum flux curve (0–200 s is speech; 201–350 s is music, and 351–450 s is environment sound)

2.5. Spectrum flux

Spectrum Flux (SF) is defined as the average variation value of spectrum between the adjacent two frames in an audio sub-clip,

$$\mathrm{SF} = \frac{1}{(N-1)(K-1)} \times \sum_{n=1}^{N-1} \sum_{k=1}^{K-1} [\log(A(n,k) + \delta) - \log(A(n-1,k) + \delta)]^2 \tag{7}$$

where

$$A(n,k) = \left| \sum_{m=-\infty}^{\infty} x(m) w(nL - m) e^{-j\frac{2\pi}{L}km} \right| \tag{8}$$

and $x(m)$ is the input discrete audio signal, $w(m)$ the window function; L is the window length; K is the order of DFT, δ a very small value to avoid calculation overflow, and N is the total frame number in one audio sub-clip.

In general, speech signals are composed of alternating voiced sounds and unvoiced sounds in the syllable rate, while music signals do not have this kind of structure. Hence, for speech signal, its spectrum flux (SF) will be in general greater than that of music, which is confirmed in our experiments. We also found that, the spectrum flux of environment sounds are among the highest and change more dramatically than those of speech and music. Based on our previous work [9], this feature is especially useful for discriminating some strong periodicity environment sounds such as tone signal, from music signals.

Figure 1 is an example of the spectrum flux for speech, music, and environment sound. The speech segment is from 0 to 200 s, the music segment is from 201 to 350 s and the environment sound is from 351 to 450 s. From the figure, we can see that SF is a good feature to discriminate among speech, environment sound and music.

2.6. Band periodicity

Band periodicity (BP) is defined as the periodicity of each sub-band. It can be derived from sub-band correlation analysis. Four sub-bands is selected again with intervals

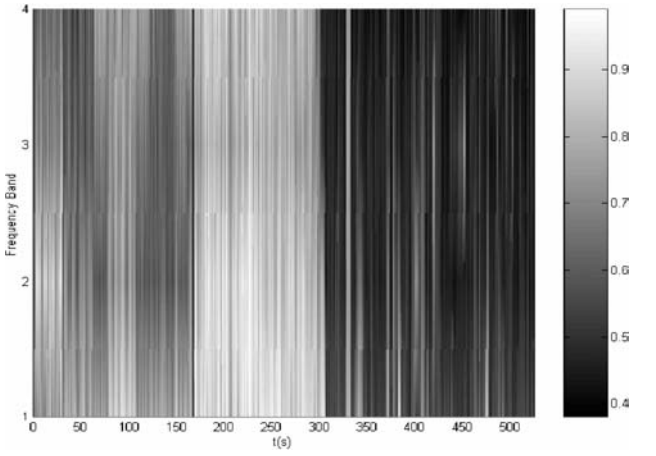

Fig. 2. Band periodicity of an example audio segment. Along the time (horizontal) axis, 0-150 s is a music segment of tube instrument, 150-300 s is piano sound, 300-520 s is the concatenation of different kinds of environment sound

$[0, \frac{w_0}{8}], [\frac{w_0}{8}, \frac{w_0}{4}], [\frac{w_0}{4}, \frac{w_0}{2}]$ and $[\frac{w_0}{2}, w_0]$. The periodicity property of each sub-band is represented by the maximum local peak of the normalized correlation function. For example, for a sine wave, its BP will be 1; but for white noise, its BP is 0. The normalized correlation function is calculated from the current frame and previous frame:

$$r_{i,j}(k) = \frac{\sum_{m=0}^{M-1} s_i(m-k)s_i(m)}{\sqrt{\sum_{m=0}^{M-1} s_i^2(m-k)}\sqrt{\sum_{m=0}^{M-1} s_i^2(m)}} \qquad (9.1)$$

where $r_{i,j}(k)$ is the normalized correlation function; i is the band index, and j is the frame index. $s_i(n)$ is the i-th sub-band digital signal of current frame and previous frame, when $n < 0$, the data is from the previous frame; otherwise, the data is from the current frame. M is the total length of a frame.

We denote the maximum local peak as $r_{i,j}(k_p)$, where k_p is the index of the maximum local peak. That is, $r_{i,j}(k_p)$ is band periodicity of the ith sub-band of the jth frame. Thus, the band periodicity is calculated as

$$bp_i = \frac{1}{N}\sum_{j=1}^{N} r_{i,j}(k_p) \quad i = 1, \ldots, 4 \qquad (9.2)$$

where bp_i is the band periodicity of ith sub-band, N is the total frame number in one audio sub-clip.

Figure 2 shows an example of band periodicity comparison between music and environment sounds. The music segment in the example is from 0 to 300 s, while the remaining part is environment sounds. It is observed that the music band periodicities are in general much higher than those of environment sound. Therefore, band periodicity is an effective feature in music and environment sound discrimination. In our real implementation, only the periodicities of the first two sub-bands are selected to discriminate different audio classes.

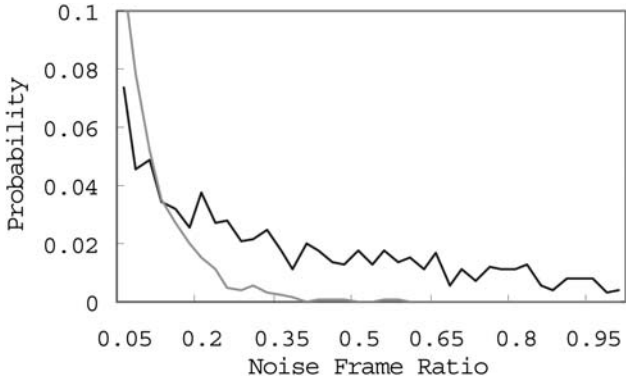

Fig. 3. The probability distribution curves of NFR; **a** music and **b** environment sound

2.7. Noise frame ratio

In order to discriminate environment sound from music and speech, and discriminate noisy speech from pure speech and music more accurately, Noise Frame Ratio is proposed.

Noise frame ratio (NFR) is defined as the ratio of noise frames in a given audio clip. A frame is considered as a noise frame if the maximum local peak of its normalized correlation function is lower than a pre-set threshold. The NFR value of noise-like environment sound is higher than that for music, because there are much more noise frames of the previous class, as illustrated in Fig. 3.

Figure 3 shows the probability distribution curves of NFR for music and environment sounds from our audio database. For music, almost no NFR value is above 0.3; however, for environment sound, the portion of NFR values that are higher than 0.3 is much higher. NFR is really depending on how noisy the signal is. The data shows some environment sound is more noise-like.

These two kind feature sets are then concatenated into combined feature vector. But the characteristics of the feature components are so different that it is not appropriate to just put these features into a feature vector. Each feature component should be normalized to make their scale similar. The normalization is processed as $x_i' = (x_i - \mu_i)/\sigma_i$, where x_i is the i-th feature component, the corresponding mean and standard derivation σ_i can be calculated from the ensemble of the training data. The normalized feature vector is considered as the final representation of an audio sub-clip.

3. Learning using support vector machines

Support vector machine (SVM) [17,18] learns an optimal separating hyper-plane from a given set of positive and negative examples. It minimizes the structural risk, that is, the probability of misclassifying yet-to-be-seen patterns for a fixed but unknown probability distribution of the data. This is in contrast to traditional pattern recognition techniques of minimizing the empirical risk, which optimize the performance on the training data. This minimum structural risk principle is equivalent to minimize an upper bound on the generalization error. A support vector machine can be either linear or nonlinear (kernel based). The former is used in linearly separable case, and the

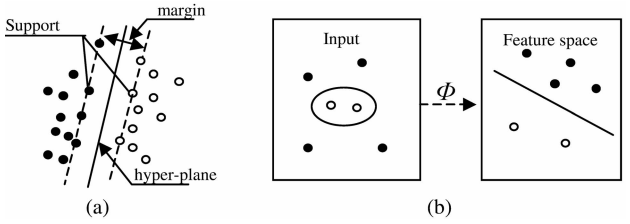

Fig. 4. a A linear SVM finds the maximum margin linear separating hyper-plane in the input space. **b** A nonlinear SVM uses a nonlinear kernel to implicitly map the data into a high dimensional feature space in which the mapped data is linearly separable

latter is used in linearly non-separable but nonlinearly (better) separable case.

The feature distribution of audio data is so complicated that different classes may have overlapping or interwoven areas. Different audio classes can not be linearly separated. However, a kernel based SVM is well suited to handle such a situation. In this section, we only introduce some concept of linear and kernel support vector machine.

3.1. Linear support vector machines

Consider the problem of separating a set of training vectors belonging to two separate classes, $(x_1; y_1), \ldots, (x_l; y_l)$, where $\boldsymbol{x}_i \in R^n$ is a feature vector and $y_i \in \{-1, +1\}$ is a class label, with a separating hyper-plane of equation $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$; of all the boundaries determined by $\boldsymbol{w}$ and b, the one that maximizes the margin (Fig. 4a) will generalize better than other possible separating hyperplanes.

On the basis of this rule, the final optimal hyper-plane classifier can be represented by the following equation:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \bar{\alpha}_i y_i \boldsymbol{x}_i \boldsymbol{x} + \bar{b}\right) \tag{10}$$

where $\bar{\alpha}$ and $\bar{b}$ are parameters for the classifier; the solution vector $\boldsymbol{x}_i$ is called as Support Vector with $\bar{\alpha}_i$ being non-zero. The detail derivation could be found in [17,18].

3.2. Kernel support vector machines

In the linearly non-separable but nonlinearly separable case, the SVM replaces the inner product $\boldsymbol{x} \cdot \boldsymbol{y}$ by a kernel function $K(\boldsymbol{x}, \boldsymbol{y})$, and then constructs an optimal separating hyper-plane in the mapped space. According to the Mercer theorem [18], the kernel function implicitly maps the input vectors into a high dimensional feature space in which the mapped data is linearly separable (Fig. 4b). This provides a way to address the curse of dimensionality [18].

Possible choices of kernel functions include: (1) Polynomial: $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^d$, where the parameter d is the degree of the polynomial; (2) Gaussian Radial Basis Function: $K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$, where the parameter σ is the width of the Gaussian function; (3) Multi-Layer perception function: $K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\kappa(\boldsymbol{x} \cdot \boldsymbol{y}) - \mu)$, where the κ and μ are the scale and offset parameters. In our method, we use the Gaussian Radial Basis kernel, because it was empirically observed to perform better than the other two.

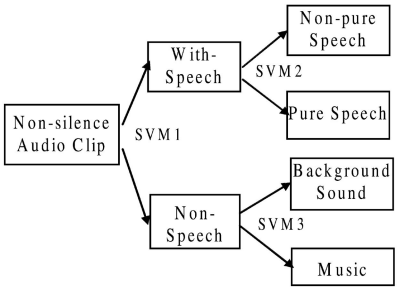

Fig. 5. Binary tree for multi-class classification

For a given kernel function, the classifier (10) is updated as,

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \bar{\alpha}_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \bar{b}\right) \tag{11}$$

4. Multi-class classification

In our work, an audio clip is classified into five classes. They are silence, music, background sound, pure speech, and non-pure speech which include speech with music and speech with noise. The input audio is first classified into silence and non-silence clip depending on the energy and zero-crossing rate information. This step is based on a rule-based classifier instead of SVM. It will be marked as silence if the energy and zero-crossing rate is less than a predefined threshold. Then, for those non-silence sub-clips, kernel SVM with Gaussian Radial Basis is used to classify the left four classes. Because support vector machine is a two-class classifier, we should figure out a multi-class classification scheme.

Classification of these classes can be achieved by combining all the two-class SVMs. There are two common schemes for this purpose: one-against-all and the one-against-one. We use a simpler scheme and construct a bottom-up binary tree for classification, as shown in Fig. 5. This is because the testing process of SVM is a little time-consuming and the binary tree strategy can reduce the number of pair-wise comparisons.

By comparison between each pair, one class number is chosen to represent the "winner" of the current two classes. The selected classes (from the lowest level of the binary tree) will come to the upper level for another round of tests. Finally, a unique class label will appear on the top of the tree.

Figure 5 illustrates the comparison process. First, the audio clip is classified into speech and non-speech classes. Then, non-speech is further classified into music and background sound, and speech clip is classified into pure speech and non-pure speech.

Obviously, it only needs 3 support vector sets to discriminate them all. In General, using this method, it only needs $c-1$ support vector (SV) sets to classify c classes, and at most it needs $\lceil \log_2 c \rceil$ times comparisons.

5. Audio segmentation

Final segmentation of an audio stream is achieved by classifying each sub-segment into an audio class. Meanwhile, as the audio stream is always continuous, it is almost impossible to change the audio types too suddenly or too frequently. Under this assumption, some smoothing rules are applied in the final segmentation of an audio sequence. More specifically, the first rule used is:

Rule 1. *if* $(s[1] \neq s[0]$ *AND* $s[2] = s[0])$ *then* $s[1] = s[0]$

where a sequence of three sub-segments is considered, $s[0], s[1], s[2]$ stands for the audio type of the last sub-segment, the current sub-segment, and the next sub-segment, respectively. This rule means, if the middle sub-segment is different from the other two but the other two is the same, then the middle one is considered as a misclassification. For example, if we detect a pattern of consecutive sub-segments like "speech-music-speech", it is most likely for the sequence to be all speeches; therefore, they will all be segmented as speech. But if we come across a sequence such as "speech-music-environment sound", the middle sub-segment can be considered either mis-classified or classified accurately. In our approach, we will uniformly rectify the middle sub-segment according to its previous one. That is:

Rule 2. *if* $(s[1] \neq s[0]$ *AND* $s[2] \neq s[0]$ *AND* $s[2] \neq s[1])$ *then* $s[1] = s[0]$

Because silence is much possible to appear in only one sub-segment, silence will not be applied by the above *Rule 1* and *Rule 2*. That is, the sequence such as "speech-silence-speech" is accepted. Thus, combining above two rules, the final rule becomes

Final Rule. *if* $(s[1] \neq s[0]$ *AND* $s[1] \neq SILENCE$ *AND* $s[2] \neq s[1])$ *then* $s[1] = s[0]$

This smoothing process can further prevent some classification errors. After the process, the segmentation results are obtained.

6. Experiments and evaluations

The database used in our experiments is composed of 2600 audio clips, which is about 4 hours in total length, collected from TV programs, the Internet, audio and music CDs with each clip labeled in terms of the pre-defined 5 classes. This database includes speech in different conditions, such as speeches in TV studios, speeches with telephone (4 kHz) bandwidth and 8 kHz bandwidth. The music content in this data set is mainly songs, and most of them are pop music. Such music contents are difficult for most of other audio classifiers. The audio data are all mono channel and 16bit per sample, but have different sample rates, including 44.1 kHz, 32 kHz, 16 kHz and 8 kHz. They are down-sampled into 8 KHz before further processing. The database is partitioned into a training set and a testing set, 1 hour for training set and 3 hours for testing set.

In our experiments, an audio stream is first divided into non-overlapping sub-clips by a sliding window. Classification is then performed on these sub-clips. If there are two audio types in a sub-clip at the same time, we classify the sub-clip into the audio type which dominates. An audio stream is thus segmented by classifying each sub-clip into one of the pre-defined five classes.

Table I. Cross validations for speech and non-speech discrimination with different training sets

Index	Training set		SVs	Testing set	
	Count	Acc.		Count	Acc.
1	3651	98.85%	407	11117	96.32%
2	3888	98.44%	500	10880	96.53%
3	3495	98.60%	423	11073	96.41%
4	3643	98.65%	415	11125	96.33%
5	3846	98.18%	437	10922	96.22%

In the experiments, we evaluated the performance of SVM-based method on audio classification, tested the effectiveness of each feature, and compared the performance between SVM and other frequently used pattern recognition method, such as k-nearest neighbor (KNN) and Gaussian Mixture Model (GMM). Three kind audio-pair classification types are used, from Fig. 5, which are speech/non-speech classification, pure speech/non-pure speech classification and music/environment sound classification.

6.1. Cross-validation

First, in order to evaluate the robustness of SVM-based audio classification method, a cross validation for speech and non-speech classification is performed with five randomly selected training sets and testing sets. The testing unit is one second in this experiment. RBF kernel is used in the classifier with parameters $\sigma = 1$. (The same configuration is used at the other experiments) The results are shown in the Table I.

In Table I, the accuracy of the training set and testing set are listed, where count means the total capacity of a training set or testing set, and SVs is the number of support vectors obtained from a training set. From Table I it can be seen that SVM-based method can perform very well. The accuracy is above 96% and differs little with the training set or the testing set. Around 12% of the training data is selected as supporting vectors; and the average accuracy (rates of correctly classified patterns for testing set) is up to 96.36%.

For other classifying types, their average discriminating accuracies are listed in Table II, which shows that high accuracy can be achieved by each discriminator. This reveals that SVM-based approach is very effective on audio classification.

6.2. Performance on audio classification and segmentation

In the above experiments, only the performance of two-type classifier is listed. For multi-class classification, another exper-

Table II. Experiment result of different classifying type

Classifying type	Average accuracy
Speech/non-speech	96.36%
Music/background sound	94.67%
Pure speech/non-pure speech	89.64%

Table III. Overall classification results before smoothing (unit: 100%)

Audio type	Total Number	Classification results			
		Pure speech	Non-pure speech	Music	Background sound
Pure Speech	100	87.63	9.97	0.66	1.74
Non-pure Speech	100	0	93.92	4.18	1.90
Music	100	0.86	3.73	91.84	3.57
Background Sound	100	3.14	8.69	5.91	82.26

Table IV. Overall segmentation results after smoothing (unit: 100%)

Audio type	Total number	Segmentation results			
		Pure speech	Non-pure speech	Music	Background sound
Pure speech	100	90.53 (+2.90)	8.30	0.26	0.91
Non-pure speech	100	0	96.20 (+2.28)	2.28	1.52
Music	100	0.53	1.85	95.45 (+3.61)	2.17
Background sound	100	1.66	6.65	4.07	87.62 (+5.36)

iment is implemented to get the overall classification results. The results are listed in Table III.

Table III showed that the system performs well. But there are still many misclassifications. Pure speech is easily misclassified into non-pure speech, while non-pure speech is not misclassified into pure-speech. Background sound and music are also easily classified into non-pure speech. This is because non-pure speech is mixed from pure-speech and music or background sound. This mixed type is easily confused with the other three pure types. Based on the continuity of audio stream, a smoothing scheme is processed for final segmentation as presented in Sect. 5. The performance has been further improved, which is shown in Table IV, in which the number in parenthesis means the corresponding improvement compared with the previous classification results.

From Table IV, it can be seen that the performance is improved and the misclassification reduced. For each audio type, the accuracy is increased by 2–5%; it is especially high for background sound. About 97% of speech can be discriminated from music and background correctly, and 90% pure-speech is classified from the other three classes. For music, only 0.53% is misclassified into pure-speech, 1.85% into non-pure-speech and 2.17% into background sound. From this data, it can be seen that the audio classification and segmentation system performs very well. This is because SVM, as a better classifier, is used in our system; and some new features are used to further improve the classification and segmentation performance. The superiority of SVM and the effectiveness of new features will be shown in the following experiments.

Table V. Effectiveness of each feature (unit: 100%)

Classifier	Baseline	Baseline $+SF$	Baseline $+BP$	Baseline $+NFR$	Baseline $+All$
Speech/ non-speech	94.17	95.48 (+1.31)	95.01 (+0.84)	95.14 (+0.97)	96.36 (+2.19)
Music/ background sound	93.08	93.50 (+1.59)	94.67 (+0.42)	94.30 (+1.59)	94.67 (+1.22)
Pure speech/ non-pure speech	87.54	89.16 (+1.62)	88.51 (+0.97)	89.16 (+1.62)	89.64 (+2.10)

6.3. Feature effectiveness

To show the effectiveness of each new feature, we first implemented a baseline classifier. The baseline system employed all commonly used features, which are listed in section 2.1–2.4. On the basis of this baseline, each new feature is added to compose a new classifier. Classification results between using the new classifier and the baseline classifier are compared to show the effectiveness of each feature. In this experiment, three classifying type are used again. The results are listed in Table V. The number in parenthesis means the corresponding improvement compared with the baseline classification results.

From Table V, it can be seen that the baseline classifier can perform well, since SVM is a good classification algorithm in audio case. However, the performance of each classifier still improves quite a lot after adding the new feature. For example, for speech and non-speech classification, the error rate is reduced by 22.5% when the feature SF is added; and the error rate is reduced by 37.6% after all three feature are used.

6.4. Performances of SVM with different testing units

In the above experiment, the testing unit is one-second subclip. That means: we could not classify the audio segment before it has one second data. One-second limit means long delay and low resolution. It could not be used in some applications which have high real-time requirements and high resolution requirements, such as real-time audio coding. In such applications, smaller classifying unit is needed. Therefore, it is necessary to evaluate the performance of SVM with smaller testing unit. It will also be a very helpful reference for selecting proper classification unit in different applications. Besides 1 second unit, five different smaller testing units are used, which are 0.5 s, 0.4 s, 0.3 s, 0.2 s, and 0.1 s, respectively. Average accuracy (which is simplified as accuracy later), recall, and precision for the positive samples are used to evaluate the performance of SVM-based audio classification method. The definitions of accuracy, recall, and precision are as the following.

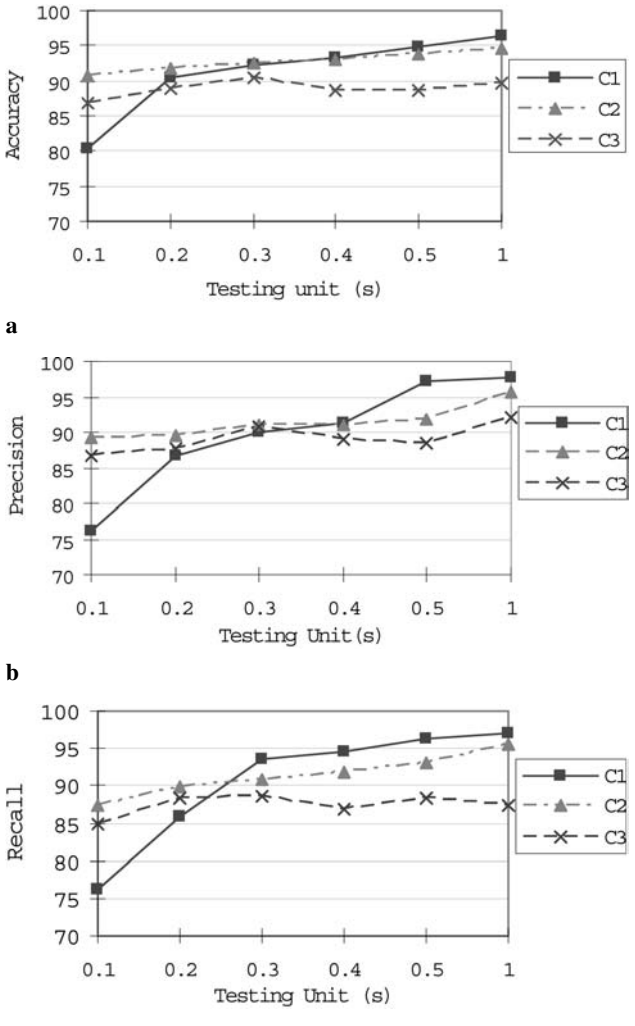

Fig. 6. a Accuracy, **b** recall and **c** precision of classification between different classes when different testing unit (s) are used, where C1 means the classification between speech and non-speech audio clip, C2 means environment sound and music discrimination and C3 means pure speech and non-pure speech discrimination

Suppose the number of positive samples is M, the number of negative samples is N in the test set; m of M positive samples and n of N negative samples are detected accurately in prediction results. Then:

$$Accuracy = \frac{m+n}{M+N} \tag{12.1}$$

$$Recall = \frac{m}{M} \tag{12.2}$$

$$Precision = \frac{m}{M+N-n} \tag{12.3}$$

The results are illustrated in the Fig. 6. Three kinds of classifications are performed on six different testing units, where C1 represents speech and non-speech discrimination, C2 environment sound and music discrimination, and C3 pure speech and non-pure speech discrimination.

It could be seen from Fig. 6 that, in general, the accuracy of classification decreases with the decrease of testing unit, although there is an exception for C3, pure speech and non-pure speech discrimination, the performance of 0.3 s is a little better than those of the other four. This is reasonable since the characteristics of each audio type are difficult to capture clearly when testing unit is very short.

For speech and non-speech discrimination, the accuracy with 0.5 s unit is of 94.95%, and the accuracy is dramatically decreased to 80.27% when the testing unit is 0.1 s. However, for music/environment discrimination and pure-speech/non-pure speech discrimination, the accuracy does not decrease dramatically. Accuracy is only decreased from 95.32% to 90.77% and 92.66% to 86.87% respectively when the testing unit is changed from 0.5 s to 0.1 s. This is because non-speech includes music and many kinds of environment sounds, and the speech set includes pure-speech, speech with music and speech with noise. When the testing unit is very short, the characteristics between music and speech with music are very similar; moreover, speech with noise is also easily classified into environment sound. It causes many misclassifications, thus the performance of classification between speech and non-speech decreases dramatically when the testing unit decreases. However, in pure-speech/non-pure-speech classification and music/environment sound classification, the audio type is relatively simple that their difference could be well captured using our efficient feature set. So the performance did not decrease so much in these two cases.

The performance seems still good when the testing unit is 0.3 s, the accuracy, precision and recall of difference classifications are all near 90%. The performance decreased sharply when the testing unit is 0.2 s or 0.1 s, especially for speech and non-speech classification. For music/environment sound classification and pure-speech/non-pure speech discrimination, the performance is above 85% even if the testing unit is 0.1 s. However, since these two classifications are after the speech/non-speech classification, leading to a low overall performance of classification.

From the experimental results, we also find that, when the classification unit is large, the accuracy of C1 is higher than those of C2 and C3; but when the classification unit is small, the accuracy of C1 will be lower than those of C2 and C3. The same attributes are shown in recall and precision.

These experimental results indicate that there is a trade-off between performance and classification unit and the trade-off should be determined based on the applications. If an application is more sensitive to accuracy, it could select a longer testing unit. If an application concerns more on real-time and time-delay, a shorter classification unit should be selected while maintaining proper accuracy.

6.5. Comparisons with KNN and GMM

We have also experimentally compared the performance of SVM with k-nearest neighbor (KNN) and Gaussian Mixture Model (GMM) on different classifications. KNN and GMM are commonly used classifier for audio classification. In Scheirer's work [5], performance differences between the classifiers and effects of parameters settings for each classifier are examined. It was found that there is very little difference between parameter setting for each classifier type. KNN-5 and GMM-20 are found to be a little bit better than other param-

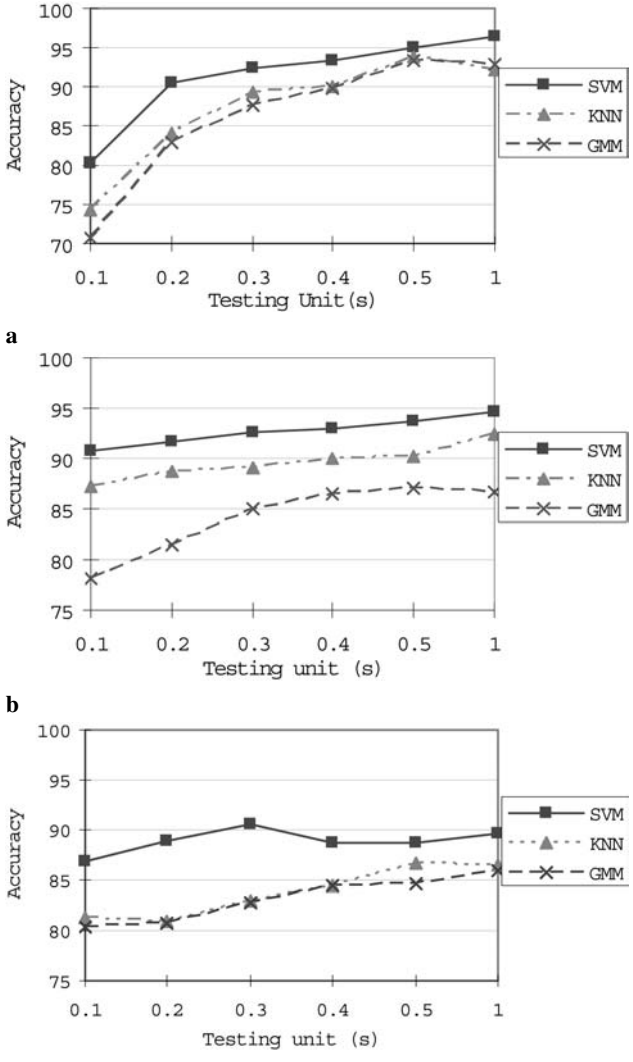

Fig. 7. Performance comparison among SVM, KNN and GMM on different classifications with different testing unit (s): **a** speech and non-speech classification; **b** environment sound and music discrimination; **c** pure speech and non-pure speech classification

eter setting. In our experiments, we also found the parameter setting has little effect on audio classification, and KNN-2 and GMM-16 are slightly better. So, in this comparison experiment, 2 neighbors are used to vote the test data into one class in K-nearest neighbor classifier, and 16 components are selected in Gaussian Mixture Model. Different classification units, which include 0.1 s, 0.2 s, 0.3 s, 0.4 s, 0.5 s and 1 s, are used again for comparison. Only accuracy is compared to evaluate the performance of SVM, KNN and GMM in this experiment. The results are shown in the Fig. 7.

From Fig. 7, it can be seen that the performance of SVM-base classifier is much better than the other two classifiers in audio classification and segmentation. In general, the performance difference between these classifiers is about 4%–8%. The smaller the testing unit is, the higher the performance difference is. So, the SVM-based classifier shows its big advantage over the other two methods, especially when the testing unit is small.

We also found that the performance of GMM-16 is a little bit lower than that of KNN-2, it is still reasonable since KNN could achieve very high accuracy if the training data is large enough. It is also seen from this experiment that whichever classifier is used, the classification performance decreases when the testing unit is deceased, especially for speech and non-speech classification. The reasons are described in the section above.

Computationally, the SVM is also more efficient than the KNN method. This is because SVM testing only depends on the support vectors obtained from the training process while KNN testing depends on the entire training set. In general, the number of support vectors is much less than the number original training data.

7. Conclusion

In this paper, we have presented in detail an SVM-based approach to classification and segmentation of audio streams. An audio clip is classified into one of the five classes: pure speech, non-pure speech, music, environment sound, and silence. Audio segmentation is performed by classifying each audio sub-segment into these five classes with a smoothing process. We have also proposed a set of features for the representation of audio streams, including band periodicity and spectrum flux. The effectiveness of these features is evaluated in experiments. Experimental evaluations show that the SVM classifier achieves high accuracy in audio classification and segmentation. It is also shown that the performance of SVM method is much higher than that of using KNN and GMM whatever the testing unit length is.

As for future direction, we will improve our classification scheme to discriminate more audio classes and refine our feature set by analyzing its redundancy and reducing its dimension. We will also focus on developing an effective scheme to apply audio content analysis to improve video structure parsing and indexing process.

References

1. J. Foote. Content-based retrieval of music and audio. In: C.C.J. Kuo et al. (eds.) Multimedia Storage and Archiving Systems II, Proc. of SPIE, volume 3229, pp. 138–147, 1997
2. J. Foote. An overview of audio information retrieval. ACM-Springer Multimedia Systems, 1998
3. S. Pfeiffer, S. Fischer, W. Effelsberg. Automatic Audio Content Analysis, Proc. of the fourth ACM international conference on Multimedia, pp. 21–30, 1996
4. J. Saunders. Real-time Discrimination of Broadcast Speech/Music. Proc. of ICASSP96, Vol. II, pp. 993–996, Atlanta, May, 1996
5. E. Scheirer, M. Slaney. Construction and Evaluation of a Robust Multifeature Music/Speech Discriminator. Proc. of ICASSP 97, vol II, pp 1331–1334, April 1997
6. D. Kimber, L. Wilcox. Acoustic Segmentation for Audio Browsers. Proc. of Interface Conference, Sydney, Australia, July, 1996
7. T. Zhang, C.-C. J. Kuo. Heuristic Approach for Generic Audio Data Segmentation and Annotation. Proc. of ACM Multimedia'99, pp. 67–76, 1999

8. S. Srinivasan, D. Petkovic, D. Ponceleon. Towards robust features for classifying audio in the CueVideo System. Proc. of the seventh ACM international conference on Multimedia'99, pp. 393–400, 1999
9. L. Lu, H. Jiang, H. J. Zhang. A Robust Audio Classification and Segmentation Method. Proc. of the 9th ACM international conference on Multimedia, pp. 203–211, 2001
10. L. Lu, Stan Li, H, J. Zhang. Content-based Audio Segmentation Using Support Vector Machines. Proc. of ICME 2001, pp 956–959, Tokyo, Japan, 2001
11. S. Z. Li. Content-based classification and retrieval of audio using the nearest feature line method. IEEE Transactions on Speech and Audio Processing, September 2000
12. E. Wold, T. Blum, D. Keislar, J. Wheaton. Content-based classification, search and retrieval of audio. IEEE Multimedia Magazine 3(3): 27–36, 1996
13. Z. Liu, J. Huang, Y. Wang, T. Chen. Audio feature extraction and analysis for scene classification. IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing, 1997
14. L. Rabiner, B. H. Juang. Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, New Jersey, 1993
15. P. J. Moreno, Ryan Rifkin. Using the Fisher Kernel Method for Web Audio Classification. Proc. of ICASSP2000, Vol. IV, pp. 2417–2420, June 2000
16. K. El-Maleh, M. Klein, G. Petrucci, P. Kabal. Speech/Music Discrimination for Multimedia Applications. Proc. of ICASSP2000, Vol. IV, pp. 2445–2448, June 2000
17. C. Cortes, V. Vapnik. Support vector networks. Machine Learning 20: 273–297, 1995
18. V. N. Vapnik. Statistical learning theory. John Wiley & Sons, New York, 1998
19. T. Joachims. Making large-scale SVM learning practical. In: B. Scholkopf, C. Burges, A. Smola (eds.) Advances in Kernel Methods – Support Vector Learning. MIT Press, 1999