CONTENT-BASED AUDIO SEGMENTATION USING SUPPORT VECTOR MACHINES

Lie Lu, Stan Z. Li and Hong-Jiang Zhang
Microsoft Research China
5/F Beijing Sigma Center, No.49 Zhichun Road
Hai Dian District, Beijing 100080, China
{i-lielu, szli, hjzhang}@microsoft.com

ABSTRACT

Audio exists at everywhere, but is often out-of-order. It is necessary to arrange them into regularized classes in order to use them more easily. It is also useful, especially in video content analysis, to segment an audio stream according to audio types. In this paper, we present our work in applying support vector machines (SVMs) in audio segmentation and classification. Five audio classes are considered: silence, music, background sound, pure speech, and non-pure speech which includes speech over music and speech over noise. A SVM learns optimal class boundaries from training data to best separate between two classes. A sound clip is segmented by classifying each sub-clip of one second into one of these five classes. Experiments on a database composed of clips of 14870 seconds in total length show that the average accuracy rate for the SVM method is much better than that of the traditional Euclidean distance based (nearest neighbor) method.

1. INTRODUCTION

Audio data is an integral part of many modern computers and multimedia applications. Numerous audio recordings are dealt with in audio and multimedia applications. Rapid increase in the amount of audio data demands for a computerized method that allows efficient and automated segmentation and classification of audio stream based on their sounds content [13, 10, 4, 5].

An important recent work is done by Wold et al. [13], in which various perceptual features (such loudness, pitch, brightness, bandwidth and harmonicity) are used to represent sound clip, and a weighted Euclidean distance and the nearest neighbor rule is used for classification. In [4], the audio representation consists of 12 Mel-frequency cepstral coefficients (MFCCs) plus energy as the audio features. A vector quantization method is used for classification. In [14], TV programs are segmented and classified, using perceptual features, into one of news-report, whether-report, commercial, basketball-game, and football-game categories. In [11], HMM is used for audio-based segmentation and classification of video, into four categories, i.e. speech, music, environmental sound and silence, among which an environmental sound is further classified into applause, explosion and bird's sound.

In this paper, support vector machines (SVMs) [1, 12] are used for the classification and segmentation of audio stream or audio clips. The reason that we chose to use a kernel SVM for the classification is the following: First, a set of training data is available and can be used to train a classifier. Second, once trained, the computation in a SVM depends on a usually small number of supporting vectors and is fast. Third, the distribution of audio data in the feature space is complicated and different classes may have overlapping or interwoven areas. A kernel based SVM is well suited to handle such a situation.

In our method, an audio clip is divided into non-overlapping one second sub-clips. These sub-clips are classified into two categories firstly, i.e. speech and non-speech; and then, speech clip is further classified into pure speech, non-pure speech; and non-speech clip is classified into background sound and music. Different support vectors are used for different class discrimination.

The rest of this paper is organized as follows. Section 2 describes how a sub-clip is represented by low level perceptual and cepstral feature. An overview of kernel SVM is surveyed in Section 3. In Section 4, a method for multi-class classification is discussed. In Section 5, experiments and evaluations on an about 4 hours-long database are showed.

2. AUDIO FEATURE SELECTION

An important step of audio classification is feature selection. Different features should be used in different methods and different applications. The most important thing is: the selected features should capture the temporal and spectral structure of different audio classes. In our approach, some new features are introduced.

In our data, all audio clips are 16-bit, mono-channel, and down-sampled into 8 KHz. They are pre-emphasized with parameter 0.98 and then divided into non-overlapping sub-clips. A sub-clip is of 1 second duration and is further divided into forty 25ms-long frames. The segmentation is performed based on the classification of these one-second sub-clips.

Two types of features are computed from each frame: (i) mel-frequency cepstral coefficients (MFCCs), and (ii) perceptual features. The mean and standard deviation of the feature trajectories over all 40 frames are considered as a feature set for this one-second sub-clip.

In our method, 8 order MFCCs are used as suggested by [9]. We also use several perceptual features such as short time energy (STE), zero crossing rates (ZCR), sub-band powers distribution, brightness, bandwidth and the pitched ratio (ratio between the number of pitched frames and the total number of frames in a sub-clip), which are often used in many other works.

In addition to these features, we also introduce some new features, which are described in detail as follows:

1. Spectrum Flux (SF), which shows the variation of spectrum between the adjacent two frames,

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

where

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

and x(m) is the input audio, w(m) the window function, L is the window length, K is the order of DFT, and δ a very small value to avoid calculation overflow.

2. Linear Spectrum Pair (LSP) divergence shape [7]. It is useful for discriminating speech and non-speech. Some LSP templates for speech are trained, then we use the minimum distance between the testing LSP and the templates as one feature of audio clip.

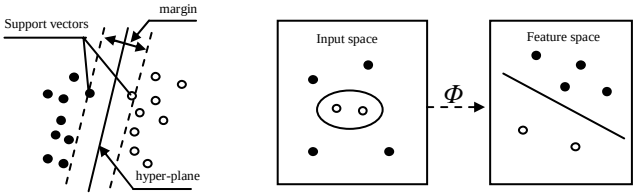
3. Band periodicity (*B*P), which is defined as the periodicity of each sub-band. It can be derived from sub-band correlation analysis. In our work, we choose four sub-bands, they are 500~1000Hz, 1000~2000Hz, 2000~3000Hz, and 3000~4000Hz respectively. The periodicity property of each sub-band can be 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.

These two feature sets are then combined as a feature vector of a frame. The mean and standard deviations of these feature vectors over all forty frames are computed, and these statistics compose a new feature vector. Finally, the feature vector is normalized by dividing each feature component by its standard deviation calculated from the ensemble of the training data. The normalized feature vector is considered as the final representation of a sub-clip.

3. LEARNING USING SUPPORT VECTOR MACHINES

3.1 Linear Support Vector Machines

Consider the problem of separating a set of training vectors belonging to two separate classes, $(\boldsymbol{x}_1; y_1),\ldots,(\boldsymbol{x}_l; y_l)$, where $\mathbf{x}_i \in R^n$ 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.1.a) will generalize better than other possible separating hyper-planes.



(a) (b)

Figure 1: (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.

A canonical hyper-plane [12] has the constraint for parameters $\boldsymbol{w}$ and b: $\min_{\mathbf{x}_i} y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] = 1$. A separating hyper-plane in canonical form must satisfy the following constraints, $y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \geq 1$, $i = 1,\ldots l$. The margin is $\frac{2}{\|\mathbf{w}\|}$ according to its definition. Hence the hyper-plane that optimally separates the data is the one that minimizes $\phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$.

The solution to the optimization problem is given by the saddle point of the Lagrange functional,

$$L(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \{y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b]-1\} \quad (3)$$

with Lagrange multipliers α_i. The solution is given by,

$$\overline{\mathbf{w}} = \sum_{i=1}^{l} \overline{\alpha}_i y_i \mathbf{x}_i \ , \quad \overline{b} = -\frac{1}{2}\overline{\mathbf{w}}\cdot[\mathbf{x}_r + \mathbf{x}_s] \quad (4)$$

where $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$ are support vectors which belong to class +1 and –1, respectively.

3.2 Kernel Support Vector Machines

In linearly non-separable but nonlinearly separable case, the SVM replaces the inner product $\mathbf{x} \cdot \mathbf{y}$ by a kernel function $K(\mathbf{x}; \mathbf{y})$, and then constructs an optimal separating hyper-plane in the mapped space. According to the Mercer theorem [12], the kernel function implicitly maps the input vectors into a high dimensional feature space (Fig.1.b). This provides a way to address the curse of dimensionality [12].

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

For a given kernel function, the classifier is given by the following equation:

$$f(\mathbf{x}) = \text{sgn}(\sum_{i=1}^{l} \bar{\alpha}_i y_i K(\mathbf{x}_i, \mathbf{x}) + \bar{b}) \quad (5)$$

4. MULTI-CLASSES CLASSIFICATION

In our work, audio is classified into five classes, they are silence, music, background sound, pure speech, non-pure speech, which includes speech over music and speech over noise.

The input audio is first classified into silence and non-silence clip, depending on the energy information. It will be marked as silence if the energy is less than a predefined threshold. And then, for those non-silence sub-clips, the left 4 classes are classified using SVM classifiers.

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 Figure 2.

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.

From figure 2, it can be seen 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.

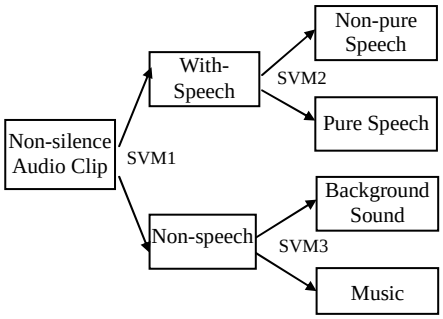


Figure 2. Binary tree for multi-class classification

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

5. EXPERIMENTS

The database used in our experiments is composed of 2610 audio clips, 14870 seconds 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. It is partitioned into a prototype (training) set of about 7450 seconds and a test set of about 7420 seconds. Five such partitioned are randomly obtained to evaluate its robustness. The results shown below are the averages of these 5 partitions if there is no specification.

In experiment, the SVM-light program of Joachims [8] is used in SVM training and classification, and RBF kernel is used with parameters σ = 1 and C = 10.

The first comparison is between the proposed SVM method and the commonly used nearest neighbor (NN) classification for speech/non-speech discrimination. For SVM method, just as Table 1 shows, around 920 supporting vectors are obtained as the result of SVM training, resulting in training error of 0.5% and testing error of 3.35%. The average accuracy (rates of correctly classified patterns for testing set) is 96.65%, significantly higher than 68.40% achieved by the NN method.

Index	Training Set		SVs	Testing Set	
	Count	Acc.		Count	Acc.
1	7578	99.49%	897	7292	96.63%
2	7407	99.55%	967	7463	97.11%
3	7638	99.35%	934	7232	96.25%
4	7347	99.62%	821	7523	96.44%
5	7287	99.49%	969	7583	96.78%

Table 1. SVM method for speech and non-speech discrimination in different training set

In Table 1, the accuracy of training set and testing set are listed, where count means the total length of training set or testing set, and SVs is the number of support vectors got from training set.

Computationally, the SVM is also more efficient than the NN method. The training of SVM takes only about 100 seconds for our training set and the testing takes less than 60 seconds for a test set.

For other classifying type, its average discriminating accuracy is listed in Table 2 in detail.

Classifying Type	Average Accuracy
Silence/non-silence	98.34%
Speech/non-speech	96.65%
Pure speech/non-pure speech	95.36%
Music/background sound	92.66%

Table 2. Experiment result of different classifying type

From Table 2, it could be noticed that high accuracy can be got for each discriminator. It shows that our approach is very effective. If we use the whole audio clip as testing unit, the accuracy will be higher, just as some works did.

For further classification, we can also divide non-pure speech into speech with noise and speech with music. This is more difficult because the statistics features of them are so similar. A potential way we can improve it is to use noise canceling algorithm before noise speech and music speech discrimination, but it is also a difficult task.

6. CONCLUSION

In this paper, we have presented in detail our approach that uses SVM for classification and segmentation of an audio clip. The proposed approach classifies audio clips into one of five classes: pure speech, non-pure speech, music, environment sound, and silence. We have also proposed a set of new features to represent a one-second sub clip, including Band Periodicity, LSP divergence shape and spectrum flux. The experimental evaluations have shown that the SVM method yields high accuracy and with high processing speed.

We are extending this work to incorporate visual information to help video content analysis, the result is also very satisfying.

7. REFERENCES

[1] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273-297, 1995.

[2] B. Feiten and S. G. unzel. "Automatic indexing of a sound database using self-organizing neural nets". *Computer Music Journal*, 18(3):53-65, 1994.

[3] B. Feiten and T. Ungvary. "Organizing sounds with neural nets". In *Proceedings 1991 International Computer Music Conference*, San Francisco, 1991.

[4] J. Foote. "Content-based retrieval of music and audio". In C. C. J. Kuo et al., editors, *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, volume 3229, pages 138-147, 1997.

[5] J. Foote. "An overview of audio information retrieval". *ACM-Springer Multimedia Systems*, 1998.

[6] S. Foster, W. Schloss, and A. J. Rockmore. "Towards an intelligent editor of digital audio: Signal processing methods". *Computer Music Journal*, 6(1):42-51, 1982.

[7] J. P. Campbell, JR. *Speaker Recognition: A Tutoria*l. Proceedings of the IEEE, vl.85, no.9, pp.1437~1462, 1997.

[8] T. Joachims. "Making large-scale SVM learning practical". In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[9] 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.

[10] Z. Liu, J. Huang, Y. Wang, and T. Chen. "Audio feature extraction and analysis for scene classification". In *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, 1997. http://vision.poly.edu:8080/paper/audio-mmsp.html .

[11] Tong Zhang and C.-C. J. Kuo. "Audio-guided audiovisual data segmentation, indexing, and retrieval". In Proc. *of SPIE Storage and Retrieval for Image and Video Databases VII*, January 1999.

[12] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[13] E. Wold, T. Blum, D. Keislar, and J. Wheaton. "Content-based classification, search and retrieval of audio". *IEEE Multimedia Magazine*, 3(3):27-36, 1996. http://muscle_sh.muscle_sh.com/ieeemm96/.

[14] Y. W. Z. Liu and T. Chen. "Audio feature extraction and analysis for scene segmentation and classification". *Journal of VLSI Signal Processing Systems*, June 1998.