A SCANNING WINDOW SCHEME BASED ON SVM TRAINING ERROR RATE FOR UNSUPERVISED AUDIO SEGMENTATION

Seyed Omid Sadjadi and John H.L. Hansen

The Center for Robust Speech Systems (CRSS),
Department of Electrical Engineering, The University of Texas at Dallas,
800 West Campbell Road, Richardson, TX 75080-3021, USA
{sadjadi, john.hansen}@utdallas.edu

ABSTRACT

Audio segmentation has applications in a variety of contexts, such as automatic broadcast news transcription, audio information retrieval, and as a pre-processing step in automatic speech recognition (ASR). The Support vector machine (SVM), as a binary classifier, is commonly used for supervised audio signal segmentation and classification. In this study, inspired by the idea of scanning window, we present and evaluate an unsupervised audio segmentation approach based on the SVM training error rate. The approach is unsupervised in the sense that it does not require prior knowledge of audio classes. Experimental results indicate that the segmentation technique outperforms traditional Bayesian information criterion (BIC), generalized likelihood ratio (GLR), and Gaussian mixture models (GMM) methods, particularly in detecting audio landmarks of short duration.

1. INTRODUCTION

Many audio streams (e.g., broadcast news from either television or radio), comprise signals from a wide variety of sources, most notably including speech and music. Since the sources are basically different in acoustic nature, a single method cannot be used to process the entire audio stream. Audio segmentation has thus become an important preprocessing step to break audio streams into homogeneous segments so that each segment can be addressed in a different manner.

State-of-the-art audio segmentation techniques include both supervised and unsupervised approaches. Supervised segmentation methods can be categorized as model-based, such as GMM or HMM [1], or decoder-based [2]. Model-based methods perform classification over a small number of frames in the audio stream, and are able to detect short duration segments. Nevertheless, these methods require pre-trained models for each audio class to be used in segmentation. They are thus limited to applications where acoustic classes are known a priori and a large amount of training data is available.

Unsupervised segmentation techniques are generally based on a likelihood ratio test between two hypotheses consisting of change and no change for a given observation sequence. Examples of these approaches include model selection based segmentation, such as Bayesian Information Criterion (BIC) [3, 4], and metric based segmentation, such as GLR [5]. These techniques, which work based on a scanning window scheme, have recently become popular because (i) they are robust and effective for the task, and (ii) they do not require prior knowledge of audio classes as models are estimated directly from the observation sequence. These facts enable them to serve a wider range of applications as well. However, both techniques detect changes over a large window, usually longer than 2 seconds, and tend to miss many short-duration segments.

Designed specifically for audio segments of short duration (i.e., less than 2 seconds), in this paper we present and evaluate an unsupervised segmentation approach, inspired by the idea of scanning window used in the above mentioned unsupervised methods and based on SVM training error rate. The approach is unsupervised in the sense that it does not require prior knowledge of audio classes.

This paper is organized as follows: In the following section, we provide a brief review of the SVM. The segmentation algorithm is described in Section 3. Section 4 presents the experiments performed, followed by a discussion of the obtained results in Section 5. Finally, we draw conclusions and discuss future work in Section 6.

2. SUPPORT VECTOR MACHINE

A SVM is a binary classifier that makes its decision by constructing an optimal separating hyperplane (OSH) that divides a d-dimensional real space into two half spaces with the largest margin [6]. Binary classification is the task of classifying the members of a given observation sequence into two groups on the basis of whether they have the same property or not. More precisely, let $\mathcal{D} = \{(\mathbf{x}_i, y_i),\ i = 0,\ \ldots,\ m-1\}$ denote a training dataset in which each example $\mathbf{x}_i \in \mathbb{R}^d$ belongs to a SVM binary class labeled as $y_i \in \{-1, +1\}$. A separating hyperplane (also called discriminant function) satisfying $\mathbf{w}^t\mathbf{x} + b = 0$, divides the dataset such that all points with the same class label are on the same side of the hyperplane, where $\mathbf{x}$ is an input vector, $\mathbf{w} \in \mathbb{R}^d$ is an adjustable weight vector, and $b \in \mathbb{R}$ is a *threshold* or *bias*. Furthermore, we let $\mathbf{w}^t$ denote the transpose of $\mathbf{w}$.

The OSH problem can be formed as,

$$\begin{cases} \text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1,\ \ i = 0,\ \ldots,\ m-1. \end{cases} \tag{1}$$

The solution to this quadratic problem can be found by computing the saddle point of the Lagrange function, where (1) is formulated as,

$$L(\alpha) = \sum_{i=0}^{m-1} \alpha_i - \frac{1}{2} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \alpha_i \alpha_j y_i y_j \mathbf{x}_i{}^t \mathbf{x}_j, \tag{2}$$

subject to,

 ISSN 2076-1465

$$\begin{cases} \sum_{i=0}^{m-1} y_i \alpha_i = 0 \\ C > \alpha_i \geq 0, \;\; i = 0, \;\ldots, \; m-1, \end{cases} \tag{3}$$

where C is a penalty parameter that determines the trade-off between margin maximization and training error minimization. Suppose that a_i maximizes (2), then, the parameter $\mathbf{w}$ of the discriminant function has the expansion,

$$\mathbf{w} = \sum_{i=0}^{m-1} a_i y_i \mathbf{x}_i. \tag{4}$$

The *bias* of the OSH can be determined from a_i and from the Karush-Kühn-Tucker (KKT) conditions as

$$b = y_j - \sum_{i=0}^{m-1} a_i y_j \mathbf{x}_i{}^t \mathbf{x}_j, \tag{5}$$

with any j such that $C > \alpha_j \geq 0$, i.e., support vectors. The corresponding training examples $(\mathbf{x}_i, y_i)$ with non-zero coefficients a_i are called support vectors. The decision function for classifying a new data point $\mathbf{x}$ can be written as,

$$f(\mathbf{x}) = \text{sgn}\left\{ \sum_{i=0}^{m-1} a_i y_i \mathbf{x}_i{}^t \mathbf{x} + b \right\}. \tag{6}$$

The decision function (6) works well when the decision boundary between the two classes is linear. However, the training set is not always linearly separable. To achieve better generalization performance, the input data can be first mapped into a high-dimensional feature space where the decision boundary is linear. As shown in Figure 1, this mapping, $\phi : \mathbf{X} \rightarrow \mathbf{F}$, can simplify the classification task. Then, the OSH is constructed in the feature space $\mathbf{F}$. If $\phi(\mathbf{x})$ denotes a mapping function that maps $\mathbf{X}$ into a high-dimensional feature space, $\mathbf{F}$, the decision function (6) becomes,

$$f(\mathbf{x}) = \text{sgn}\left\{ \sum_{i=0}^{m-1} a_i y_i G(\mathbf{x}_i, \mathbf{x}) + b \right\}, \tag{7}$$

where $G(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ is called the kernel function and must be a positive-definite function [7]. Examples of such positive-definite functions are as follows,

$$\text{Linear kernel} \quad \longrightarrow \; G(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j,$$

$$\text{Polynomial kernel} \quad \longrightarrow \; G(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^n,$$

$$\text{Gaussian RBF kernel} \quad \longrightarrow \; G(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right),$$

where $(\cdot)$ denotes the dot product, $n \in \mathbb{N}$ is the degree of the polynomial kernel, and $\sigma \in \mathbb{R}$ is the width of the Gaussian radial basis function (RBF) kernel. In addition to the above mentioned kernels, there are other kernels which are not exploited in this study (for more details see [6]).

3. SEGMENTATION ALGORITHM

Using the SVM or generally kernel-based techniques for the task of audio segmentation is not a novel concept. Lu *et al.* [2] adopted a bottom-up binary tree combining three two-class SVM classifiers for content-based audio segmentation. Ramona and Richard [8] presented a SVM-based approach

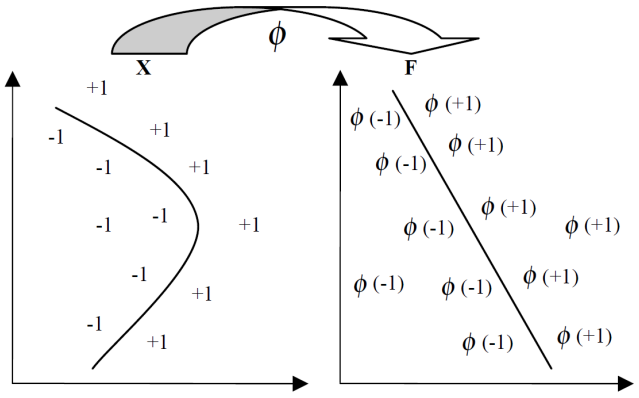

Figure 1: Feature map can simplify the classification task.

combined with a median filter post-processing for the task of speech/music segmentation. Both of these methods, are similar to model based segmentation methods, since they require a large amount of training data as well as pre-determined audio classes to train the SVM classifier. In other words, their methods are supervised. In an unsupervised framework, Lin *et al.* [9] introduced a novel speaker change detection approach based on the SVM called SVM training misclassification rate (STMR). The foundation of our algorithm is taken from their work. Other unsupervised kernel-based approaches for audio segmentation have also been proposed in [10, 11]. Although we also use the SVM, this study presents an unsupervised method for the task of audio segmentation.

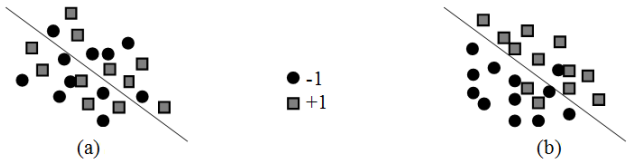

Figure 2: 2-D SVM hyperplane for classifying audio features in two adjacent windows when they come (a) from the same class, and (b) from different classes.

The basic concept of the segmentation algorithm is illustrated in Figure 2. After framing and feature extraction, an audio stream is represented as a sequence of frames with d-dimensional features. Next, for training the SVM classifier, two windows which comprise the same number of frames are considered. Inspired by the idea of window scanning which has been widely used in conventional unsupervised segmentation methods such as BIC and GLR, we begin with the assumption that there is a change point located in the audio stream at the center of the two adjacent windows under consideration. The data of these two windows, separated by this hypothetical change point, are labeled as $(+1)$ and (-1) for training the SVM hyperplane. As shown in Figure 2(a), if these two windows come from the same class, they will not have significant differences, and the SVM hyperplane will not be able to effectively discriminate between them. On the contrary, if the two windows come from different classes, they will have significant differences so that the SVM hyperplane can effectively classify these data into two classes (see Figure 2(b)).

Hence, there will be a relatively large training error rate when the two windows are from the same class while it will be small for different classes. If this training error rate is below a threshold value, the hypothesized change point will be accepted, otherwise it will be rejected. After one hypothesized change point has been tested, the two adjacent windows are moved one frame to the right, and then the new hypothesized change point will be tested again. This procedure is repeated until the right hand window reaches the end of the audio stream. This technique provides independent hyperplane training and also training error computation for every two adjacent windows, thus we can avoid the error broadcasting problem [9] that is an important drawback of methods such as BIC [3]. The segmentation algorithm is summarized as follows:

1. Construct feature set $\mathbf{X} = \{\mathbf{x}_i,\ i = 0,\ \ldots,\ m-1\}$ from the frames of the audio stream (where i is the frame index).
2. Label frames of the left and right hand windows of the hypothesized change point r as (-1) and $(+1)$ respectively, i.e., $\{\mathbf{x}_i,\ i = 0,\ \ldots,\ r-1\} \in (-1)$ and $\{\mathbf{x}_i,\ i = r,\ \ldots,\ m-1\} \in (+1)$, where $r = \dfrac{m}{2}$.
3. Train an SVM classifier using these labeled frames.
4. Test the SVM classifier with the same data used for training and calculate the training error rate.
5. If the training error rate is below a threshold, accept r as a change point, else disregard it.

4. EXPERIMENTS

4.1 Database

The problem of speech/music discrimination [12, 13, 14, 8] has attracted significant research effort for more than a decade, motivated primarily because it is essential for automatic transcription of broadcast news as well as audio information retrieval [15, 16]. Considering this, we conducted our experiments on Scheirer and Slaney's database [13] which comprises both speech and music segments recorded at random times from FM radio[1]. The total duration is 40 minutes consisting of 20 minutes of speech from both male and female speakers, as well as 20 minutes of music including samples of classical, jazz, pop, rap, and rock music, with and without vocals.

Figure 3 shows the distribution of duration of the audio segments derived manually from the database. About 33% of the segments are less than 2 seconds in order to enable us to assess our claim that the segmentation algorithm is capable of detecting changes in short duration segments. These segment are concatenated to form audio streams for performing experiments.

In our experiments, 13-dimensional mel-frequency cepstral coefficients (MFCCs) are extracted every 10 ms from frames of 20 ms duration as audio features. While originally developed for ASR applications, the MFCCs have been shown to be quite useful for music modeling, and in particular for speech/music discrimination [17].

[1] http://www.ee.columbia.edu/~dpwe/sounds/musp/music-speech-20051006.tgz

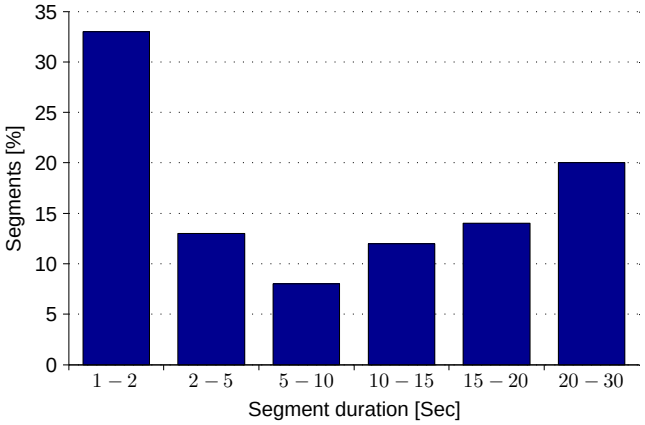

Figure 3: Distribution of audio segments durations in the database constructed by concatenating the segments manually derived from Scheirer and Slaney's database.

4.2 Evaluation Metrics

In an audio segmentation system, two possible errors can occur. Type-I errors occur if a true change is not detected within a certain neighborhood (0.25 second in our case). Type-II errors occur if a detected change does not correspond to a true change (also called false alarm). Type I and II errors can be measured in terms of precision (PRC) and recall (RCL) respectively, which are defined as,

$$\text{PRC} = \frac{\text{no. of correctly found changes}}{\text{total no. of changes found}}, \tag{8}$$

$$\text{RCL} = \frac{\text{no. of correctly found changes}}{\text{total no. of true changes}}. \tag{9}$$

Recall is usually stressed more than precision in evaluating segmentation algorithms since false alarms can be compensated by subsequent procedures such as clustering or classification [18]. Here, both metrics are treated equally in this study.

The F-measure combines PRC and RCL into one measure as,

$$\text{F-measure} = \frac{2 \times \text{PRC} \times \text{RCL}}{\text{PRC+RCL}}. \tag{10}$$

The F-measure takes on values between 0 and 1, where a higher score on this metric indicates better performance. In our experiments, in a parameter tuning stage, the threshold values and parameters for each segmentation algorithm are chosen to maximize the F-measure.

5. RESULTS

In this section, the performance of the proposed approach for speech/music segmentation is compared against that of conventional segmentation techniques based on the metrics introduced in the previous section.

We first evaluate the segmentation algorithm on audio data using the SVM with different kernel functions as mentioned in Section 2. Figure 4 shows an example of this experiment on a 9-second audio stream with 3 change points. We assume there is only one specific speaker or music genre

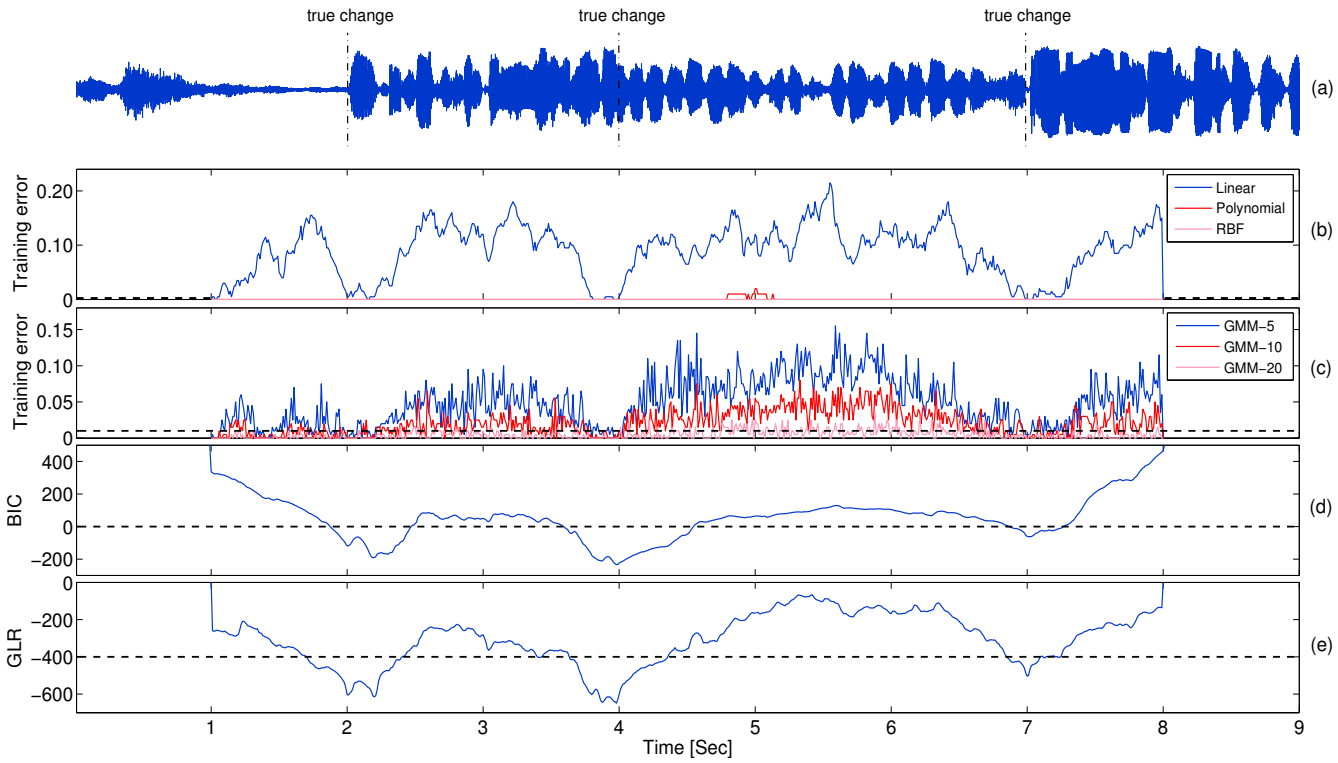

Figure 4: The audio stream (a), SVM training error rate trajectories for different kernel functions with $n = 2$ for the polynomial, $\sigma = 1$ for the Gaussian RBF, and $C = 1$ for all kernels (b), GMM training error rate trajectories for different number of Gaussians $(5, 10, 20)$ (c), BIC curve with $\lambda = 2.75$ (d), and GLR curve (e). Dashed and dash-dotted lines represent thresholds and true change points, respectively.

in each audio segment (this was considered while concatenating the audio segments). Two adjacent windows are chosen to be 100 frames (1 second) and are shifted one frame to the right in iterations of the algorithm until the right hand window reaches the end of the audio stream. Also, the parameter C of the SVM classifier is set to 1 in all iterations of our experiments. As previously noted, in the parameter tuning stage, these parameters are adjusted to maximize the F-measure. Experiments in the parameter tuning stage are conducted on ten 9-second audio streams which are randomly selected from the database. The best parameter settings obtained from this stage are used for the following experiments.

As can be seen from Figure 4(b), the SVM classifier along with the polynomial kernel ($n = 2$) or the Gaussian RBF kernel ($\sigma = 1$), always correctly classifies frames of the two adjacent windows into two classes. Consequently, training error rate trajectories for these two kernels are almost zero over the time and this phenomenon causes false alarms to occur for any threshold. On the other hand, the trajectory for the SVM trained with the linear kernel drops to nearly zero only around change boundaries with no false alarms occurring elsewhere in the figure. Therefore, the algorithm can work well with the linear kernel.

To support our claim that the proposed algorithm is superior in situations of short segment duration (or equivalently small amount of training data), we replace the SVM classifier with a GMM classifier and perform the same procedure in the segmentation algorithm to calculate the training error rate trajectories. The performance of the GMM with a varying number of Gaussians is illustrated in Figure 4(c). It is seen that as the number of Gaussians increases, the false alarm rate increases. Also, like the SVM with the Gaussian RBF kernel, the GMM with 20 Gaussians always has a training error of nearly zero. Furthermore, enormous fluctuations of GMM trajectories make it difficult to obtain a reliable threshold, so a low-pass filtering is required to eliminate redundant local minima by smoothing the trajectories.

To assess performance of other unsupervised methods and compare them with ours, we perform the same experiment on both the BIC and GLR. The results for the BIC with a penalty parameter $\lambda = 2.75$ (obtained during the parameter tuning stage) and the GLR are shown in Figures 4(d) and 4(e), respectively. The main issue with these methods is that their local minima in different change points vary in magnitude which makes the threshold setting more difficult than for the SVM. In addition, the BIC tends to miss some change points in the stream. Moreover, the GLR produces some false alarms, while the proposed algorithm maintains reliable and error free detection performance. We assume these phenomena are the result of an insufficient amount of data in short segments which prevent distance calculations from being accurate in both the BIC and GLR methods.

One of the most important reasons why our algorithm is able to detect short segments is that the SVM classifier requires only a small amount of training data in comparison to the BIC and GLR that require much more data to enable them to detect changes accurately.

Table 1 shows performance evaluation metric scores for the proposed technique using the SVM along with the linear kernel as well as the GMM with 5 Gaussians. Also

given in the table are scores obtained from evaluating the BIC ($\lambda = 2.75$) and GLR segmentation algorithms. Threshold settings for these four methods are 0, 0.01, 0, and -400, respectively, which have been obtained from the parameter tuning stage (also shown in Figure 4 as dashed lines). From a total of 40 minutes audio data, 1.5 minutes (3.75%) were used for parameter tuning, while the remaining 38.5 minutes (96.25%) have been used as test data for benchmarking the segmentation techniques. It is worth mentioning here that no smoothing was applied to the trajectories during the evaluations. In each column of the table, PRC, RCL and F-scores are reported for short segments (less than 2 seconds) and longer ones separately. Obtained results indicate that the proposed technique outperforms other conventional segmentation methods in both short and long duration segments.

Table 1: Results obtained from evaluating the segmentation techniques on approximately 40 mintues test data.

Method	**PRC**		**RCL**		**F-measure**	
	$\leq 2s$	$> 2s$	$\leq 2s$	$> 2s$	$\leq 2s$	$> 2s$
SVM	**98.7**	**96.1**	**97.4**	**98.9**	**98.1**	**97.5**
GMM	85.3	88.0	82.5	82.1	83.9	85.0
BIC	77.2	84.4	72.8	79.0	74.9	81.6
GLR	78.7	83.8	79.2	81.0	79.0	82.4

6. CONCLUSIONS

This study has presented an unsupervised audio segmentation algorithm, inspired by the idea of scanning window used in metric-based approaches, and based on the SVM training error rate. Based on the experimental results obtained in terms of PRC, RCL, and F-measure, the algorithm consistently outperformed conventional methods such as the BIC, GLR and GMM, in accurate detection of change points in audio streams. In particular, the segmentation algorithm can identify audio content changes with less audio data, making it capable of detecting landmarks of short duration (less than 2 seconds). This work can be expanded by considering more classes than speech and music. In addition, the integration of this algorithm into a framework which includes a classification step can also be considered.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers who provided constructive comments and suggestions.

REFERENCES

[1] M. Rajapakse and L. Wyse, "Generic audio classification using a hybrid model based on GMMs and HMMs," in *Proc. 11th Int'l Multimedia Modeling Conf., MMM 2005*, Melbourne, Australia, January 2005, pp. 53–58.

[2] L. Lu, S. Li and H.-J. Zhang, "Content-based audio segmentation using support vector machines," in *Proc. IEEE ICME'01*, Tokyo, Japan, August 2001, pp. 956–959.

[3] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, February 1998, pp. 127–132.

[4] P. Delacourt and C. J. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Commun.*, vol. 32, no. 1, pp. 111–126, September 2000.

[5] Q. Jin and T. Schultz, "Speaker segmentation and clustering in meetings," in *Proc. INTERSPEECH'04*, Jeju Island, Korea, October 2004, pp. 597–600.

[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.

[7] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. PAMI*, vol. 27, no. 5, pp. 801–805, May 2005.

[8] M. Ramona and G. Richard, "Comparison of different strategies for a SVM-based audio segmentation," in *Proc. EUSIPCO'09*, Glasgow, Scotland, August 2009, pp. 20–24.

[9] P.-C. Lin, J.-C. Wang, J.-F. Wang, and H.-C. Sung, "Unsupervised speaker change detection using SVM training misclassification rate," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1234–1244, September 2007.

[10] B. Fergani, M. Davey, and A. Houacine, "Unsupervised speaker indexing using one-class support vector machines," in *Proc. EUSIPCO'06*, Florence, Italy, September 2006.

[11] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, and O. Cappe, "A regularized kernel-based approach to unsupervised audio segmentation," in *Proc. IEEE ICASSP'09*, Taipei, Taiwan, April 2009, pp. 1665–1668.

[12] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proc. IEEE ICASSP'96*, Atlanta, GA, USA, May 1996, vol. 2, pp. 993–996.

[13] E. Schreier and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. IEEE ICASSP'97*, Munich, Germany, April 1997, vol. 2, pp. 1331–1334.

[14] G. Richard, M. Ramona, and S. Essid, "Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams," in *Proc. IEEE ICASSP'07*, Honolulu, HI, USA, April 2007, vol. 2, pp. 461–464.

[15] J.H.L. Hansen *et al.*, "Advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Trans. ASP*, vol. 13, no. 5, pp. 712–730, September 2005.

[16] R. Huang and J.H.L. Hansen, "Advances in unsupervised audio classification and segmentation for the Broadcast News and NGSW Corpora," *IEEE Trans. ASLP*, vol. 14, no. 3, pp. 907–919, May 2006.

[17] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int'l Symp. Music Inf. Retrieval, ISMIR'00*, Plymouth, MA, October 2000, pp. 11–23.

[18] S.O. Sadjadi, S.M. Ahadi, and O. Hazrati, "Unsupervised speech/music classification using one-class support vector machines," in *Proc. 6th Int'l Conf. Inf. Commun. Signal Process., ICICS'07*, Singapore, December 2007, pp. 1–5.