A DYNAMIC PROGRAMMING APPROACH TO SPEECH/MUSIC DISCRIMINATION OF RADIO RECORDINGS

Aggelos Pikrakis, Theodoros Giannakopoulos and Sergios Theodoridis

Dept. of Informatics and Telecommunications, University of Athens, Greece
e-mail: {pikrakis, tyiannak, stheodor}@di.uoa.gr, URL: http://www.di.uoa.gr/dsp

ABSTRACT

This paper treats speech/music discrimination of radio recordings as a maximization task, where the solution is obtained by means of dynamic programming. The proposed method seeks the sequence of segments and respective class labels (i.e., speech/music) that maximize the product of posterior class label probabilities, given the within the segments data. To this end, a Bayesian Network combiner is embedded as a posterior probability estimator. Tests have been performed using a large set of radio recordings with several music genres. The experiments show that the proposed scheme leads to an overall performance of 92.32%*. Experiments are also reported on a genre basis and a comparison with existing methods is given.*

1. INTRODUCTION

Speech/Music discrimination refers to the problem of segmenting an audio stream and labeling each segment as either speech or music. Since the first attempts in the mid 90's, a number of speech / music discrimination systems have been proposed in various application fields.

In [1], a real-time technique for speech/music discrimination was proposed, focusing on the automatic monitoring of radio stations, using features related to the short-term energy and zero-crossing rate (ZCR). In [2], thirteen audio features were used in order to train different types of multidimensional classifiers, such as a Gaussian MAP estimator and a nearest neighbor classifier. In [3], energy, ZCR and fundamental frequency were used as features in order to achieve analysis of on-line audiovisual data. Segmentation/classification was achieved by means of a procedure based on heuristic rules. A framework based on a combination of standard Hidden Markov Models and Multilayer Perceptrons (MLP) was used in [4] for speech/music discrimination of broadcast news. An Adaboost - based algorithm, applied on the spectrogram of the audio samples, was used in [5] for frame-level discrimination of speech and music. In [6], energy and ZCR were employed as features and classification was achieved by means of a set of heuristic criteria in an attempt to exploit the nature of speech and music signals.

The majority of the previously described methods deal with the problem of speech/music discrimination in two separate steps: first, the audio signal is split into segments by detecting abrupt changes in the signal statistics and at a second step the extracted segments are classified as speech or music by using standard classification schemes. The work in [4] differs in the sense that the two tasks are performed jointly by means of a standard HMM, where, for each state, a MLP is used as an estimator of the continuous observation densities required by the HMM.

The method that we propose in this paper formulates speech/music discrimination as a maximization task. In other words, the method seeks the sequence of segments and the respective class labels (i.e., speech/music) that maximizes the product of posterior (class label) probabilities, given the segments data. In order to estimate the required posterior probabilities, a Bayesian Network (BN) Combiner is trained and used. Since an exhaustive approach to this solution is unrealistic, we resort to dynamic programming to solve this maximization task.

Section 2 describes the feature extraction stage. Section 2.1 formulates speech/music discrimination as a maximization task and provides a dynamic programming solution. The BN combiner architecture and related issues are given in Section 2.2. The datasets that we have used, the method's performance (both average and on a radio genre basis), as well as a comparison with other approaches are presented in Section 3.

2. FEATURE EXTRACTION

At a first step, the audio recording is broken into a sequence of non-overlapping short-term frames and five audio features are extracted per frame. At the end of this feature extraction stage, the audio recording is represented by a sequence $\mathbf{F}$ of five-dimensional feature vectors, i.e., $\mathbf{F} = \{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_T\}$, where T is the number of short-term frames. The specific choice of features was the result of extensive experimentation. It must be emphasized that this is not an optimal feature set in any sense, and other choices may also be applicable. If $\{x(0), x(1), \ldots, x(N-1)\}$ is the set of samples of a short-term frame, then the adopted features are given by:

1. **Short-term Energy:** This is a popular feature, defined by the equation $E = \frac{1}{N}\sum_{n=0}^{N-1} x^2(n)$.
2. **Chroma-Vector based features:** The chroma vector has been widely used in various music information retrieval applications, e.g., [7]. It can be computed from the magnitude of the DFT of each short-term window, if the DFT coefficients are grouped into 12 bins, where each bin represents one of the 12 pitch classes of western-type music (semitone spacing). In this paper, two sequences of chroma vectors are extracted, using different mid-term window sizes. Each chroma sequence serves as the basis to compute a feature value over time, namely:
   Chroma-based Feature 1: The audio stream is parsed with a non-overlapping mid-term window of length 100msecs. For each frame, the chroma vector is extracted and the *standard deviation* of its twelve coefficients is computed, yielding a one-dimensional feature over time. Our study revealed that the mean value of this feature is

distinctly lower for music segments than for speech segments.

Chroma-based Feature 2: The audio stream is mid-term processed with a non-overlapping window that is 200ms long. Each mid-term window is then broken into shorter non-overlapping frames, each of which is 25msecs long, resulting into 8 short-term frames per mid-term window. At a next step, the chroma vector is extracted from each short-term frame, yielding 8 chroma vectors per mid-term widow. In order to extract the second chroma-based feature, the standard deviation of each vector coefficient is computed over the short-term frames of a mid-term window and the *minimum deviation* value is kept as the feature value. This stems from our observation (after careful experimentation) that, in music segments, there is at least one chroma coefficient with low standard deviation for a short period of time. On the other hand, in speech segments, the standard deviation of *all* chroma coefficients is high.

3. **The first two Mel Frequency Cepstral Coefficients (MFCCs)**. The filter bank used for the computation of the MFCCs consists of 40 triangular bandpass filters, with bandwidth and spacing determined by a constant mel-frequency interval. More specifically, the first 13 filters are linearly-spaced with 133.33Hz between center frequencies and are followed by 27 log-spaced filters, whose filter centers are separated by a factor of 1.0711703 in frequency. The adopted filter bank covers the frequency range $0-8$KHz, suggesting a sampling rate of $16KHz$. If $\widetilde{S_k}$, $k=1,\ldots,40$ is the output of the k-th filter, then the first two MFCCs are given by the equation

$$\widetilde{c_n} = \sum_{k=1}^{40} (\log \widetilde{S_k}) \cos[n(k-\frac{1}{2})\frac{\pi}{40}], \;\; n=1,2$$

The above suggests that the chroma-based features are computed with a different sampling rate compared with the short-term energy and the first two MFCCs. This is not a restriction, as it will be made clear in section 2.2, where the feature sequences are fed as input to a Bayesian Network combiner that serves as the posterior probability estimator.

2.1 Speech/Music discrimination treated as a maximization problem

In this stage, speech/music discrimination is treated as a maximization task, where the solution is obtained by means of dynamic programming. We make two assumptions concerning the length of the segments to be formed: a) a segment has to be at least T_{min} frames long and b) its duration cannot exceed T_{max} frames. The minimum segment duration is detected by the nature of the signals under study, i.e., we assume that a segment must have sufficient duration (0.5secs in this paper) in order to be interpreted either as speech or music. The need for T_{max} (3secs in this paper) will be made clear in the rest of this section and it is a common assumption in such problems, i.e., in variable duration HMMs. As a result, any segment longer than T_{max}, will be partitioned in segments of smaller than T_{max} length.

To proceed further, certain definitions are first given. Let L be the length of a feature sequence $\mathbf{F}$ that has been extracted from an audio stream. Our goal is twofold: a) Segment the sequence into K segments and b) classify each one of the segments as speech or music. Let $\{d_1, d_2, \ldots, d_{K-1}, d_K\}$ be the frame indexes that mark the end of each segment. Clearly, $T_{min} \leq d_1 < d_2 \ldots < d_K = L$ and $T_{max} \geq d_k - d_{k-1} \geq T_{min}$, $k = 2,\ldots,K$. Therefore, the k-th segment starts at frame index $d_{k-1}+1$ and ends at frame index d_k, with the exception of the first segment, that starts at the first frame and ends at frame index d_1 (initialization step). Thus, the feature sequence, $\mathbf{F}$, yields the following sequence of pairs

$$\{(1,d_1),(d_1+1,d_2),\ldots,(d_{K-1}+1,L)\},$$

where each pair holds the frame indexes of the beginning and end of the corresponding segment. In addition, let c_k be the class label of the k-th segment, where c_k can be either speech or music. To this end, let $p(c_k \mid \{O_{d_{k-1}+1},\ldots,O_{d_k}\})$, be the posterior probability of class label c_k given the sequence of observations (feature sequence) of the k-th segment.

Following the above notation, for any given sequence of K segments and corresponding class labels, we form the cost function

$$J(\{d_1,d_2,\ldots,d_{K-1},d_K\},\{c_1,c_2,\ldots,c_{K-1},c_K\}) \equiv$$
$$p(c_1 \mid \{O_1,\ldots,O_{d_1}\}) \prod_{k=2}^{K} p(c_k \mid \{O_{d_{k-1}+1},\ldots,O_{d_k}\}) \quad (1)$$

where independence between successive segments has been assumed. It is now possible to treat speech/music discrimination as a maximization problem. In other words, we seek the optimal sequence of segments (i.e., start and end point of each segment) and the corresponding class labels that maximize J. Equivalently, J needs to be maximized over all possible values of K, $\{d_1,d_2,\ldots,d_{K-1},d_K\}$ and $\{c_1,c_2,\ldots,c_{K-1},c_K\}$, under the two assumptions made in the beginning of this section. Obviously, an exhaustive approach would amount to an excessive computational load. Thus, we resort to dynamic programming to obtain a solution to the problem in an efficient way. Note that this is the first time that the segmentation classification task is cast in such a formulation.

To this end, as it is common with dynamic programming techniques [8, 9], we first construct a grid by placing the feature sequence on the x-axis and the two states (speech/music) on the y-axis. This is shown in Figure 1, where S stands for speech and M stands for music. Clearly, the grid has two rows and L columns (L being the length of the feature sequence). In order to give a physical meaning to the nodes of the grid, take, as an example, node (O_{d_k},S), $T_{min} \leq d_k \leq L$. This node stands for the case that a speech segment ends at frame index d_k. Following this line of reasoning, a path of K nodes $\{(O_{d_1},c_1),(O_{d_2},c_2),\ldots,(O_{d_K},c_K)\}$, corresponds to a possible sequence of segments, where $T_{min} \leq d_1 < d_2 < d_k = L$, $T_{max} \geq d_k - d_{k-1} \geq T_{min}$, $k = 2,\ldots,K$ and $\{c_1,\ldots,c_K\}$ are the respective class labels. We denote the transition to node (O_{d_k},c_k) from its predecessor in the path, i.e., $(O_{d_{k-1}},c_{k-1})$, by $(O_{d_{k-1}},c_{k-1}) \rightarrow (O_{d_k},c_k)$. This transition can be interpreted as follows: a segment with class label c_{k-1} ends at frame d_{k-1} and the next segment in the sequence starts at frame $d_{k-1}+1$, ends at frame d_k and has class label c_k. We then define a cost function $T(.)$ for the transition $(O_{d_{k-1}},c_{k-1}) \rightarrow (O_{d_k},c_k)$ as follows:

$$T((O_{d_{k-1}},c_{k-1}) \rightarrow (O_{d_k},c_k)) = p(c_k \mid \{O_{d_{k-1}+1},\ldots,O_{d_k}\}) \quad (2)$$

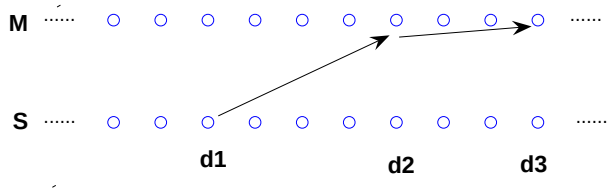

Figure 1: Dynamic programming grid.

In other words, the cost of the transition is set equal to the posterior probability of the class label, c_k, given the feature sequence defining the segment, $\{O_{d_{k-1}+1},\ldots,O_{d_k}\}$. Equation 2 holds for all nodes in the path, except for the first node (which does not have a predecessor). For the first node, $p(c_1 \mid \{O_1,\ldots,O_{d_1}\})$, stands for the posterior probability of class label c_1 given the first d_1 observations.

Equations 1 and 2 suggest that, for a given sequence of K nodes (segments) and corresponding class labels the cost function now becomes

$$p(c_1 \mid \{O_1,\ldots,O_{d_1}\}) \prod_{k=2}^{K} T((O_{d_{k-1}},c_{k-1}) \rightarrow (O_{d_k},c_k)) = J(\{d_1,d_2,\ldots,d_{K-1},d_K\},\{c_1,c_2,\ldots,c_{K-1},c_K\}) \quad (3)$$

According to equation 3, the value of function $J(.)$ for a sequence of segments and corresponding class labels can be equivalently computed as the cost of the respective path of nodes in the grid. Therefore, the optimal segmentation sequence, i.e., the sequence that maximizes $J(.)$ can be also treated as a best path sequence on the grid.

In order to compute the best-path sequence, we need to define how the best predecessor of each node in the grid is chosen. We first turn our attention to the case where a node, (O_{d_k},c_k) is not the first node in a path ($k \neq 1$). In this case the node has to be reached from a node (O_{d_l},c_l) such that $T_{min} \leq d_l < d_k$ and $T_{min} \leq d_k - d_l \leq T_{max}$. Following Bellman's optimality principle, if $J(\{d_1,d_2,\ldots,d_l\},\{c_1,c_2,\ldots,c_l\})$ is the cost of the best path up to node (O_{d_l},c_l), then the best predecessor of node (O_{d_k},c_k) is the one that maximizes the product $J(\{d_1,d_2,\ldots,d_l\},\{c_1,c_2,\ldots,c_l\})T((O_{d_l},c_l) \rightarrow (O_{d_k},c_k))$. If (O_{d_1},c_1) is the first in the path, i.e., $T_{min} \leq d_1 \leq T_{max}$ we also need to compute $p(c_1 \mid \{O_1,\ldots,O_{d_1}\})$ and take into account these values while deciding for the node's predecessor. This procedure is repeated for all nodes in the grid and the coordinates of the predecessor for each node are stored. In the end, we turn our attention to the last column of the grid and choose the node with maximum value as the winner. The winning node will be the last node of the best path. Then, we backtrack through the chain of predecessors to reveal the best path.

As it will be presented in the next section, we have chosen to approximate $p(c_k \mid \{O_{d_{k-1}+1},\ldots,O_{d_k}\})$ by means of a Bayesian Network combiner. This justifies the need to set a maximum segment duration (3secs in our study), because the BN does not yield reliable estimates when the segment's length exceeds T_{max}.

2.2 Bayesian Network architecture

As it was stated in Section 2.1, a BN has been used as a posterior probability estimator in the problem definition. To this end, the BN is trained as a classifier for the binary classification problem of speech versus music. In other words, *given a segment*, the BN is designed as a classifier combiner *that returns the posterior probability* (on which the class label is decided). It is important to emphasize, that this classifier structure decides upon the segment as a whole. This led us to derive features that are statistics computed over the whole length of the segment. Furthermore, such a choice does not require any assumption for independence among the observations within a segment. This is a very important feature offered by the use of Bayesian networks as joint probability estimators. The classifier system consists of individual, simple classifiers, that are combined by a BN architecture.

2.2.1 Individual Classifiers

At a first step, given a segment, a separate statistic is calculated for five different features. The statistics that we use are shown in Table 1. The choice of the statistics was a result of extensive experimentation and was motivated by the nature of the audio signals under study. Each one of the statistics

Feature	Statistic
Energy	$\frac{\sigma^2}{\mu^2}$
Chroma 1	μ
Chroma 2	$\frac{max}{\mu}$
MFCC 2	σ^2
MFCC 1	μ

Table 1: Statistics for each one of the five features.

is fed as input to an individual single thresholding classifier, which takes a binary decision, i.e., decides whether the feature statistic has originated from a speech or music segment. The individual decisions are then combined using a BN, which makes the final decision, as described in 2.2.2.

2.2.2 Bayesian Network Combiner

The idea behind such a procedure is to use very simple (one dimensional) classifiers, and then use a BN as a combiner to boost the overall performance. As already mentioned, the use of a BN as a final combiner is that it is a natural choice as a probability estimator (which after all is our goal).

BNs are directed acyclic graphs (DAGs) *that encode conditional probabilities* among a set of random variables. Each node of the graph corresponds to a separate random variable and the arcs of the graph encode the probabilistic dependence of the random variables (nodes). In the case of discrete random variables, for each node (random variable) A, with parents B_1, ..., B_k a conditional probability table (CPT) $P(A|B_1,...,B_k)$, is defined. In this paper, the BN architecture shown in Figure 2 ([10]), has been used as a scheme for combining the decisions of the individual classifiers described in 2.2.1. We will refer to this type of BN as the BNC (Bayesian Network Combiner). Nodes h_1, ..., h_n (also called hypotheses, rules, attributes or clauses) correspond to the binary decisions of the individual classifiers, while node Y is the output node and corresponds to the true class label. During the BN training stage, one has to learn the CPTs of the BN according

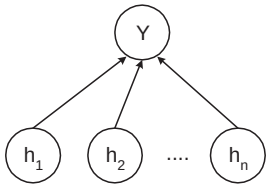

Figure 2: BNC architecture.

to the set:

$$S = \{(h_1(1),\ldots,h_n(1),s(1)),\ldots,(h_1(m),\ldots,h_n(m),s(m)))\} \quad (4)$$

where $h_j(i)$ is the result of the classifier $j = 1,\ldots,n$, for input x_i^j, where x_i^j is the feature value presented to the j-th classifier, representing the i-th input pattern, $s(i)$ is the *true label* for $x_i^j, j = 1,\ldots,n$ and m is the total number of training samples. Set S is generated by validating each individual classifier with a test set of length m. In our case a set of m audio segments, with known true class label, were used for the training. The CPTs of the BN are learned according to the Maximum Likelihood principle ([11]).

The BN is designed to make the final decision, based on the conditional probability $P_{dec} = P(Y|h_1,...,h_n)$. The process of calculating P_{dec} is called *inference* and it is, in general, a very time consuming task. However, for the adopted BNC architecture no actual inference algorithm is needed, since the required conditional probability is given directly by the CPT. Another advantage of the specific architecture *is that no assumption of conditional independence among the input nodes is made* [11].

To summarize, in the current work, a BN trained as a binary classifier. This conditional classification probability is computed in a three-step process, namely:

1. For any segment, the values of the five statistics are calculated, i.e., x_j, $j = 1,\ldots,5$.
2. x_j is fed as input to the j-th classifier. Therefore, five binary decisions h_j are extracted.
3. $P_{dec} = P(Y|h_1,...,h_5)$ is calculated by inferring in the trained BN.

3. EXPERIMENTS - RESULTS

3.1 Data Sets

The following data sets were collected from several Internet radio stations and cover a wide range of speakers and radio genres. All recordings were monophonic with a 16KHz sampling rate.

1. D_1: For creating this data set, 170 minutes of recordings were manually segmented and labelled as music or speech. This resulted in 1100 homogeneous segments of duration of 0.50 to 5.0 seconds. D_1 was used for training and testing the Bayesian Network classifier. Thus, our BNC network has been trained for all possible segment lengths that may occur during the DP optimization.
2. D_2: This data set consists of uninterrupted audio recordings from distinct radio broadcasts (more than 10 hours of total duration) and was used for testing the proposed segmentation scheme. To this end, the recordings were also manually segmented and labeled. Furthermore, D_2 was divided into 7 subsets according to radio genre (e.g. news, rock, etc), in order to test performance on a genre basis as well.

3.2 BN-related training and testing issues

In order to train and test the Bayesian Network Classifier, data set D_1 has been used. In particular, 20% of the audio segments of D_1 were used for testing the BNC, along with the individual classifiers. The results of the classification performances of the individual classifiers and the BNC are displayed in Table 2. The best individual classifier (in terms of error rate) is the one based on the 1st MFCC. The error reduction of the combination scheme compared to the error of the best classifier is $e_{red} = 100\frac{|e_{best}-e_{bnc}|}{e_{best}} \simeq 36\%$. The boosting in performance achieved by the Bayesian Network as a classifier combination scheme is obvious. The 3.5% performance justifies our decision of using the BNC as posterior probability estimator, given the segment.

	Music	Speech	Overall
Energy	21%	13%	17%
Chroma#1	5%	9%	7%
Chroma#2	8.5%	8.5%	8.5%
MFCC#1	7.5%	3.5%	5.5%
MFCC#2	14.5%	10.5%	12.5%
BNC	3.5%	3.5%	3.5%

Table 2: Error rates (%) of the individual classifiers and the BNC.

3.3 Performance of the proposed method

The experiments were carried out for 7 separate radio genres. Genre names and respective recording durations are presented in Table 3. Beside the Confusion Matrices for each genre, the overall accuracy of each segmentation scheme was calculated, along with the music and speech precision and the music/speech recall. Each element $C_{i,j}$ of the confusion matrix corresponds to the percentage of data whose true class label was i and was classified to class j. From C, one can directly extract the recall and precision values for each class:

Radio Genre	Duration (minutes)
Pop-Rock	125
Jazz-Blues	90
Dance	85
News	80
H. Metal - H. Rock	80
Rap - RnB	75
Classical	75

Table 3: Radio genres and respective recording durations.

1. **Recall** (R_i). R_i is the proportion of data with true class label i, that were correctly classified in that class. For example, the recall of music is calculated as $R_1 = \frac{C_{1,1}}{C_{1,1}+C_{1,2}}$.
2. **Precision** (P_i). P_i is the proportion of data classified as class i, whose true class label is indeed i. Therefore, music precision is $P_1 = \frac{C_{1,1}}{C_{1,1}+C_{2,1}}$.

The results are displayed in Tables 4 and 5.

An implementation of the proposed system is publicly available on the Internet at `http://www.di.uoa.gr/sp_mu`. In terms of response times, the implemented system is comparable with other approaches in the literature (e.g., [4]). It has to be noted that the publicly available version also includes a pre-processing and a boundary correction (post-processing) stage. The use of these two extra stages results in improved overall accuracy (exceeds 94.5%) without increasing response times. Due to space restrictions, the description of these two stages has been omitted in this paper.

	Precision		**Recall**		
	Mus.	Sp.	Mus.	Sp.	**Overall**
Pop-Rock	96.0	95.8	99.3	80.0	96.0
Jazz-Blues	99.0	92.6	96.2	98.0	96.8
Dance	87.9	78.0	95.2	56.6	86.2
News	75.4	99.4	97.0	93.9	94.4
Heavy Metal	99.1	86.2	99.1	85.3	98.3
Rap-RnB	94.5	34.8	84.3	62.8	81.8
Classical	93.6	96.6	99.3	74.9	94.1

Table 4: Discrimination results (%) per radio genre.

	Music	**Speech**
Music	69.24%	2.83%
Speech	4.85%	23.08%

Table 5: Average Confusion Matrix of the proposed method, computed over all examined genres. *Overall Accuracy: 92.32%*.

3.4 Comparison with other methods

This section is an attempt to compare the proposed scheme against methods that have been presented in the literature by other authors. Such a comparison turned out to be a difficult task due to the diversity of data sets that have been used in the literature and the inherent difficulties in reproducing other authors' work. As a result, we have chosen to summarize in this section the key performance issues of selected papers as presented by the respective authors. It has to be noted that the dataset in this paper is significantly larger than datasets studied by other authors. In addition, in this paper an attempt is made to present results per music genre, which is not the case in any of the other papers. More specifically :
[4]: Results are reported for four artificially created datasets (40 minutes total audio duration). The reported performance varies in the range 93% – 96%. The origin of datasets poses an inherent difficulty in comparing this method with other approaches in the literature.
[5]: Works on a frame-level basis. A binary (speech/music) classification decision is taken separately for each short-term frame. The dataset consists of 240 audio recordings, each of which is 15secs long (total recording duration is 1 hour). Part of the dataset is used for training purpses. An accuracy of 88%, on frames sampled at 20msec intervals, is reported. When a smoothing technique is applied, the performance rate reaches 93%.
[6]: The total speech duration in the audio corpus was 3 hours and 9 minutes, which was subdivided by the segmentation algorithm into about 800 segments (over-segmentation); 97% of these segments were correctly classified as speech. The total music duration in the audio corpus was 52 min, which was subdivided by the segmentation algorithm into about 400 segments (over-segmentation); 92% of these segments were correctly classified as music.

4. CONCLUSION

A novel speech - music discrimination technique is proposed in the current paper. The performance of the proposed system was evaluated using real radio data covering a wide range of music genres. The average accuracy of the method is 92.32%, whereas for specific music genres (e.g., Hard Rock-Heavy Metal) the accuracy of the system reaches 98%.

REFERENCES

[1] J. Saunders, "Real-time discrimination of broadcast speech/music", *Proc. ICASSP 1996*, vol 2, pages 993-996, Atlanta, May 1996.

[2] E. Scheirer and M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator", *Proc. ICASSP 1997*, pages 1331–1334, Munich, Germany.

[3] Tong Zhang and C.-C. Jay Kuo, "Audio Content Analysis for Online Audiovisual Data Segmentation and Classification", *IEEE Transactions On Speech And Audio Processing*, Vol. 9, No. 4, May 2001.

[4] Jitendra Ajmera, Iain McCowan and Herve Bourlard, "Speech/music segmentation using entropy and dynamism features in a HMM classification framework", *Speech Communication*, vol.40, 2003, pp. 351-363.

[5] N. Casagrande, D. Eck, and B. Kigl. "Frame-level audio feature extraction using AdaBoost.", in *Proc. ISMIR 2005*, London, UK, 2005.

[6] C. Panagiotakis and G. Tziritas, "A Speech/Music Discriminator Based on RMS and Zero-Crossings", *IEEE Trans. Multimedia*, vol. 7(1), pp. 155–166, Feb. 2005.

[7] Mark A. Bartsch and Gregory H. Wakefield, "Audio Thumbnailing of Popular Music Using Chroma-Based Representations", *IEEE Transactions Multimedia*, Vol. 7, No. 1, February 2005.

[8] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 3d edition*. Academic Press, 2005.

[9] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in *Proceedings of the IEEE*, Vol. 77, No. 2, 1989

[10] A. Garg, V. Pavlovic and T.S. Huang, "Bayesian Networks as Ensemble of Classifiers", in *Proc. of the IEEE International Conference on Pattern Recognition*, pp. 779-784, Quebec City, Canada, August 2002.

[11] D. Heckerman, "A Tutorial on Learning With Bayesian Networks", Microsoft Research, MSR-TR-95-06, Mar. 1995.