A SPEECH/MUSIC DISCRIMINATOR FOR RADIO RECORDINGS USING BAYESIAN NETWORKS

Theodoros Giannakopoulos, Aggelos Pikrakis and Sergios Theodoridis

Dept. of Informatics and Telecommunications
University of Athens
Panepistimioupolis, 15784, Athens, Greece
{tyiannak,pikrakis,stheodor}@di.uoa.gr

ABSTRACT

This paper presents a speech/music discriminator for radio recordings. The segmentation stage is based on the detection of changes in the energy distribution of the audio signal. For the classification stage, Bayesian Networks have been adopted in order to combine the results of nine k-Nearest Neighbor classifiers trained on individual features. To this end, a comparison of the performance of three popular Bayesian Network architectures is presented. Furthermore, in order to reduce the number of features used for classification, a new feature selection scheme is introduced, that is also based on the properties of Bayesian Networks. The proposed system has been tested on real Internet broadcasts of BBC radio stations.

1. INTRODUCTION

The task of automatic discrimination of speech and music is very important as a preprocessing stage in many audio content characterization applications, such as musical genre classification and speech (speaker) recognition. There have been several speech/music discrimination systems proposed during the last years. In [1], a real-time speech/music discriminator for the automatic monitoring of radio channels was proposed, based on energy contour and zero crossing rate (ZCR). In [2], thirteen audio features have been used to train four different types of classifiers, including a multidimensional Gaussian MAP estimator, a spatial partitioning scheme based on k-d trees and a nearest neighbor classifier. In [3], ZCR and line spectral frequencies (LSFs) have been used for frame-level speech music discrimination. In [4], [5] Hidden Markov Models (HMMs) have been employed as speech/music classification tools.

This paper presents: (a) a segmentation algorithm based on the detection of changes in the RMS distribution of the signal (b) a classification scheme based on a Bayesian Network (BN) that combines the outputs of nine individual k-Nearest Neighborh classifiers, each of which has been trained on a separate feature and (c) a new feature selection scheme based on Bayesian Networks. The proposed system has been tested on real Internet broadcasts of BBC radio stations. The next section describes the proposed segmentation scheme. Section 3 presents the feature extraction and classification schemes. The last section reports on the experiments that have been carried out. Finally, conclusions are given in Section 5.

2. SEGMENTATION ALGORITHM

At a first stage, the audio stream is divided into non-overlapping segments by means of a segmentation algorithm. We have adopted an "incremental" approach to audio segmentation. A segment grows from a "seed", one step at a time. In other words, each time a fixed number of audio signal samples is added at the end of the segment, provided that certain criteria related to the RMS distribution are satisfied. More specifically:

Step 1: The algorithm starts by assigning the first T seconds of the signal to an initial segment. T is a user-driven parameter, set equal to 1 second for our experiments.

Step 2: For this segment, the RMS sequence is calculated by splitting the segment into non-overlapping short-term windows (each window is 10ms long). The distribution of the resulting RMS sequence is approximated by a generalized χ^2 distribution, defined by the pdf $p(x) = \frac{x^a e^{-bx}}{b^{a+1}\Gamma(a+1)}, x \geq 0$. Parameters a and b are related to the mean and variance of the RMS sequence, $a = \frac{\mu^2}{\sigma^2} - 1$ and $b = \frac{\sigma^2}{\mu}$.

Step 3: The segment is extended by attaching T more seconds of audio to its end. The new RMS sequence is computed and let a_1 and b_1 be the new values for the χ^2 distribution. If $\mid a - a1 \mid \leq 0.05a$ and $\mid b - b1 \mid \leq 0.05b$, the segment's extension is approved, the segment is replaced by the extended one and Step 3 is repeated by attaching T more seconds of audio at a time. If it happens that at least one of the two criteria does not hold, the segment's extension is not approved, the segment's time boundaries are stored, and the T seconds of the signal that served as the candidate extension form the "seed" of a new segment. Then the procedure starts again from Step 2 for this new segment.

The philosophy behind this approach is that, when an abrupt change occurs in the signal, e.g. a transition from speech to music (and vice versa), the change is reflected in the shape of the histogram of the RMS sequence of the extended segment, and this change of shape will result in significantly modified values for the a and b parameters. We assume that any change larger that 5% suggests an abrupt transition in the signal, hence the two criteria that we adopted. Our approach bares certain similarities with region growing techniques in image segmentation, where regions develop from "seeds" by attaching neighboring pixels, provided that the resulting regions remain homogeneous according to specified criteria.

The previous χ^2 distribution test was also suggested in [6], yet the method there is different. In [6], neighboring segments of predefined length are statistically compared (local decision), while in our method segments are built incrementally. This allows the system to base its decisions over larger time periods, which, however, are not

fixed but data adaptive.

After the segmentation stage is complete, the length of each segment is examined. Segments longer than a pre-defined threshold (2 seconds for our experiments) are preserved and are fed to the classifier. Segments shorter than the threshold are marked as candidates for further processing. If two or more such segments are neighbors in terms of time, they are merged to form a single segment. The rationale behind this hybrid segmentation/merging scheme is that long-segments (i.e., longer than 2 seconds), usually contain pure music or pure speech from a single speaker (who is likely to utter words clearly and with a steady rate). Short segments are most likely the result of over segmentation, due to the presence of rapid changes in speaking rates. Despite this over-segmentation, it makes sense to treat such consecutive short-length segments as a group, merge them into one larger segment and feed the resulting segment to the classifier.

For the above segmentation scheme to work effectively, segments are not allowed to grow arbitrarily long. A segment's extension is halted if it reaches a predefined length. This was chosen after extensive experimentation to be 10 seconds for our system. This is because, if an abrupt transition, say from music to speech, takes place in the signal and the music segment is, for example 30 seconds long, adding speech to its end will not alter significantly the distribution of the RMS sequence, because the histogram will be biased toward music. Therefore, the abrupt change will only manifest itself in the histogram after a number of speech extensions are added, thus resulting into poor segmentation boundaries.

3. AUDIO CLASSIFICATION

We have adopted an approach based on BNs for the classification of segments as speech or music. Toward this end, a set of nine features is extracted from each segment. For each feature, a separate classifier is trained and the individual classifiers are combined using a BN. Hence, each classifier operates in a different feature space. In this way we "increase" the independence between individual classifiers, which is desirable in order to combine classifiers [7]. The importance of each feature for classification purposes is investigated by a novel feature selection scheme, based on a new probabilistic criterion that stems from the nature of BNs.

3.1. Feature Extraction

Nine commonly used features are extracted from each audio segment, namely: **Spectral Centroid**, **Spectral Flux**, **Spectral Rolloff**, **Zero Crossing Rate**, **Frame Energy** and four **Mel-frequency cepstral coefficients** (MFCCs) ([7]). For this purpose, a short-term moving window is applied on each segment. The length of the moving window depends on the particular feature (20ms for the first four features, 40ms for energy and 50ms for the MFFCs). For all features, a 50% overlap has been adopted for successive windows.

At a second step, a number of statistics are extracted for the above features on a segment basis. For this purpose, each segment is split into non-overlapping sub-segments (0.5secs long). For each sub-segment, the statistics shown in Table 1 are extracted for each feature. These are then averaged over all sub-segments in order to generate a two-dimensional feature vector for the whole segment. As can be seen from Table 1, the adopted statistics include central moments and three ratios, namely the maximum value to median value ratio and the percentage of short-term frames whose value is higher/lower than a predefined threshold (multiple of the median value). The choice of statistics was motivated by the nature of the signals under study. For example, concerning energy, the number of short-term frames whose energy value is lower than 10% of the median value, is higher for speech segments, because they are likely to consist of more silent periods than music segments. The specific choice of statistics for each feature, was the result of extensive experimentation.

Feature	1st Statistic	2nd Statistic
Sp. Centroid	max/median	3rd central mom.
Sp. Flux	standard dev.	3rd central mom.
Sp. Rolloff	standard dev.	3rd central mom.
ZCR	max/median	3rd central mom.
Fr. Energy	# frames$\geq$3med	# frames$\leq$0.1med
MFCCs	standard dev.	4th central mom.

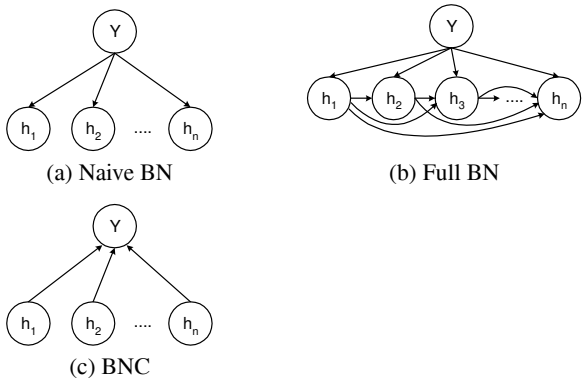
Table 1.

3.2. Classification Scheme

Each one of the above feature vectors is fed as input to an individual k-Nearest-Neighbor (kNN) classifier, which takes a binary decision, i.e., decides whether the feature has originated from a speech or music segment. The individual decisions are then combined using a BN, which makes the final decision. The BN architectures used are described next.

3.2.1. BN Topologies

In order to proceed, let us first define a number of terms related to BNs. BNs are directed acyclic graphs (DAGs) that encode conditional probabilities between a set of 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. Figure 1 presents three popular BN architectures, which have been adopted and compared in this paper. In all architectures, the output of n classifiers, h_1, ..., h_n is fed as input to a BN ($Y, h_1, ..., h_n \in \{0, 1\}$), whose output node makes the final combined decision, based on the conditional probability $P_{dec} = P(Y|h_1, ..., h_n)$. Nodes h_1, ..., h_n are also called hypotheses, rules, attributes or clauses. The process of calculating the output probability is called *inference*.



(a) Naive BN (b) Full BN

(c) BNC

Fig. 1. Three BN acrhitectures examined in this paper.

We have used three different BNs (shown in Figure 1). The first of these is also known as a Naive BN, the second as a fully-connected BN and we will refer to the third one as a BNC. The reason that we

experimented with three different BN architectures is that each one of them exhibits certain advantages and drawbacks. For example, although naive BNs are easy to implement, train and understand, they rely on the assumption that all clauses are independent given the class value [8]. On the other hand, the fully-connected BN [9] captures all dependencies between attributes, but inference in such a model has been proved to be an NP-complete problem [7]. The BNC was proposed in [10] and makes no assumption of conditional independence between the individual classifiers' results. For computing the probability P_{dec} of the Naive BN, we used Pearl's algorithm ([11]). In the fully-connected BN, Loopy Belief Propagation (LBP) is used. Finally, for the BNC structure, there is no need of applying any inference algorithm, since the required conditional probability is extracted directly from the CPT.

3.2.2. BN Training

All BNs are trained using the set

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

where $h_j(i)$ is the result of classifier $j = 1, \ldots, 9$ for input vector $\underline{x}_i^j$, where $\underline{x}_i^j$ is the feature vector presented to the j-th classifier representing the i-th input pattern, $s(i)$ is the *true label* for $\underline{x}_i^j, j = 1, \ldots, 9$ and m is the total number of training samples. In order to generate S, each individual classifier is validated with a test set of length m. The CPTs of the BN are learned according to the Maximum Likelihood principle ([12]). In our application, in order to generate S, we must first test each individual classifier with a set of m audio segments whose true class label is known.

It may happen that some distributions that populate the CPTs of the BN during the training stage, are absent from the training set. This will result in a 0.5 output probability while using the BN as a classifier of unknown segments. In the current work, we propose two alternative approaches for dealing with this phenomenon, namely: either label the segment as *"Unclassified"*, or use a *a majority voting* rule for classification.

3.3. Automatic Feature Selection

In order to reduce the number of features, we propose a new BN approach to feature selection. To this end, we first construct the BN of figure 2, to which we refer as *BNerror*. The nodes of this BN correspond to the binary variables e_i, $i = 1, \ldots, 9$, which take the values $e_i = 0,\ if\ h_i(x) = s(x)$ and 1 otherwise. In other words, $e_i = 0$, if the i-th classifier has generated a correct decision. The training set for the BNerror is directly extracted from S. After training, BNerror can be used to calculate, per individual classifier, the value of a criterion. Let us take for example feature $i = 1$, which is associated with h_1. Then the respective criterion becomes

$$C(1) = \sum_{e_2} \ldots \sum_{e_9} \{P(e_1 \mid e_2, \ldots, e_9) \sum_{j=2}^{9} e_j\} \quad (2)$$

$C(i), i = 2, \ldots, 9$ are similarly defined. In words, $C(i)$ measures the mean conditional probability that h_i classifies correctly a segment given the results of all other classifiers, multiplied by the number of classifiers that misclassified that pattern. This weighted conditional probability is actually a measure of accuracy of each classifier emphasizing on the cases of failure of other classifiers. In other words, if $C(i)$ is high, classifier h_i is generally accurate when other classifiers fail. We select the classifiers with the highest values of C. In order to calculate the probabilities in (2), we use LBP to infer in BNerror.

4. EXPERIMENTS

We created two separate datasets from five BBC Internet radio stations, covering a wide range of speakers and musical genres (monophonic recordings, 16KHz sampling rate). For the first dataset (D_1), 170 minutes of recording were manually segmented and labeled as music or speech. The second dataset (D_2) consisted of three uninterrupted audio recordings from three distinct radio broadcasts (3 hours of total recording duration).

4.1. BN comparison results

D_1, being free of segmentation errors, was used to determine the best BN architecture, whereas D_2 was used to measure the overall accuracy of the system by applying the best BN scheme. To this end, a random part of D_1 (20%) was used to train the individual kNN classifiers, 40% to generate the training set for the BNs and the remaining 40% to calculate the classification accuracy of the BNs. Table 2 presents the error rates of the individual classifiers and the BN combiners. Figure 3 demonstrates the error reduction achieved by the BNs compared to the best single classifier. This reduction is drawn as a function of the size of the training set for the BNs. The error reduction, e_{red}, is defined as: $e_{red} = 100 \cdot (e_{min} - e_{comb})/e_{min}$, where e_{comb} is the combiner's error rate and e_{min} is the minimum error rate of the individual classifiers. The following conclusions were extracted: **(a)** The feature with the smallest classification error is the 1st MFCC coefficient, **(b)** The BNs error is always smaller than 9%, **(c)** The BNs classification accuracy increases with the size of the training set, **(d)** The Naive BN is more accurate than the other two combiners, only for very small training sets (less than 2000 training samples), **(e)** Full BN and BNC exhibit comparable error rates. Taking into account the above conclusions, we have chosen the BNC combiner, because in terms of computational complexity, is the least demanding scheme (one order of magnitude faster than the Naive BN and two orders of magnitude faster than the fully-connected BN).

4.2. Overall system assessment

The final set of experiments refers to the overall assessment of the system, involving both the automatic segmentation and classification stages. The next stage of experiments involved segmentation scheme along with the BNC combiner that was previously elected, in order to determine the overall accuracy of the system, i.e. including segmentation errors. The BNC architecture along with the individual classifiers were re-trained using D_1 (20% for the individual classifiers and the remaining 80% the BNC). We segmented the 3 audio recordings of D_2 (3 hours of recording time), using the segmentation algorithm described in Section 2, and then applied BNC on the

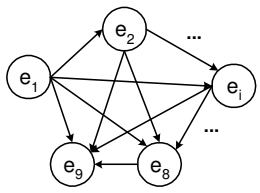


Fig. 2. The BNerror structure

	Segment length (secs)				
Classifier	**1**	**2**	**5**	**10**	**15**
Centroid	18.5	17.1	14.6	13.5	10.6
Flux	31.6	27.8	17.1	15.3	14.4
Rolloff	20.4	20.3	15.3	13.0	10.6
ZCR	19.7	20.6	16.3	14.7	10.3
Energy	25.6	18.6	14.8	13.1	11.6
MFCC 1	10.8	10.0	7.7	7.4	4.0
MFCC 2	16.4	15.7	13.6	12.1	10.0
MFCC 3	12.5	11.5	8.8	8.0	4.1
MFCC 4	15.8	15.5	11.8	11.0	6.4
Naive BN	8.8	8.6	6.9	6.4	3.4
Full BN	8.4	8.3	6.7	6.7	3.7
BNC	8.5	8.2	6.8	6.6	3.5

Table 2. Error rates for $D1$.

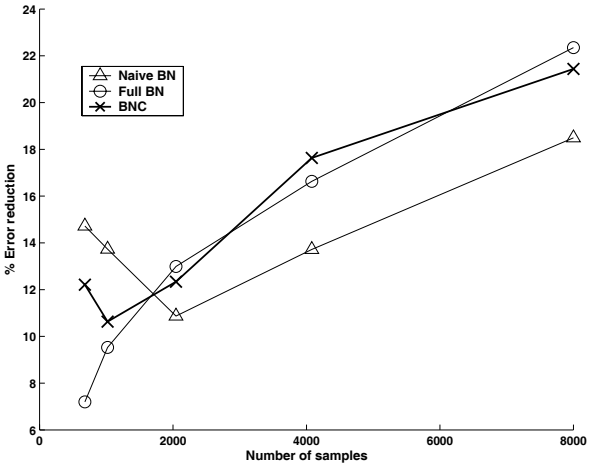


Fig. 3. %Error reduction for different number of samples

extracted segments, using the "best" five classifiers (according to criterion C), that correspond to the following audio features: 3rd, 1st and 4th MFCCs, Spectral Cetnroid and Frame Energy. In average, the overall system error rate using BNC is 5.48%, while the error rate using only the best classifier is 6.34%. This corresponds to 13.5% error reduction. These results were extracted using the majority voting rule as an alternative classifier when the BNC produces a 0.5 output probability (as explained at the end of Section 3.2.2).

The system was also tested for the alternative of leaving unclassified segments, when the combiner's output has a probability of 0.5. The results are presented in table 3. The column labeled "NC" displays the percentage of the data that has not been classified.

Sequence	**% error**	**% N.C.**
BBC 4	2.93	4.03
BBC 5	5.77	0.72
BBC 6	3.99	1.24
Average	4.23	1.99

Table 3. Error rate and % of not classified audio data

5. CONLCUSIONS

This paper presented a speech/music discriminator based on BNs. A new segmentation scheme was employed and three different BN architectures were compared. A novel feature selection technique, based on BNs has been proposed. This technique is general and can be applied to any pattern recognition problem. The system was tested on real Internet radio broadcasts and achieved an average classification accuracy of 94.5%. Taking into account that the experiments were conducted on real-life data, this is a very promising result.

6. REFERENCES

[1] J. Saunders, "Real-time discrimination of broadcast speech/music," in *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Atlanta, May 1996, vol. 2, pp. 993–996.

[2] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Munich, Germany, 1997, pp. 1331–1334.

[3] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Orlando, FL, USA, 2000, vol. 1, pp. 2445–2448.

[4] J. Ajmera, I. McCowan, and H. Bourlard, "Robust hmm based speech/music segmentation," in *In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Orlando, FL, USA, May 2002, vol. 1, pp. 297–300.

[5] J. Ajmera, I. McCowan, and H. Bourlard, "Speech/music segmentation using entropy and dynamism features in a hmm classification framework," *Speech Communication*, vol. 40, no. 3, 2003.

[6] C. Panagiotakis and G. Tziritas, "A speech/music discriminator based on rms and zero-crossings," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 155–166, Feb. 2005.

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

[8] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, 1997.

[9] J. Davis, V. Santos Costa, I.M. Ong, D. Page, and I. Dutra, "Using bayesian classifiers to combine rules," *In Proceedings of the 3rd SIGKDD Workshop on Multi-Relational Data Mining*, 2004.

[10] A. Garg, V. Pavlovic, and T.S. Huang, "Bayesian networks as ensemble of classifiers," *In Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 779–784, 2002.

[11] J. Pearl, "Fusion, propagation and structuring in belief networks," *Artificial Intelligence*, vol. 29, no. 3, pp. 241–288, 1986.

[12] D. Heckerman, "A tutorial on learning with bayesian networks," *Microsoft Research, MSR-TR-95-06*, Mar. 1995.