Speech/Music Discrimination Using Discrete Hidden Markov Models

Karnebäck, S.

journal:	TMH-QPSR
volume:	46
number:	1
year:	2004
pages:	041-059

Speech/Music Discrimination Using Discrete Hidden Markov Models

Stefan Karnebäck
stefan@speech.kth.se
Centre for Speech Technology, KTH - Royal Institute of Technology,
Lindstedtsvägen 24, SE-100 44 Stockholm, Sweden

Abstract

A speech/music discrimination system using discrete Hidden Markov Models has been designed. The system has been evaluated using separate training, development and test databases. The discrimination ability was examined using different features and feature combinations and results are presented as error rate on the development and test databases. Features were chosen from knowledge about the speech signal. Adding the zero crossing rate, autocorrelation function and spectral gravity features to cepstrum coefficients helped to improve the discrimination result, while the cepstrum features were found to be more robust. The impact of the model size on the Speech/Music discrimination result was especially evaluated. Different compositions of the database were also explored, with and without a good match.

The best result on a mismatched situation, 2.3% error rate on test data, was achieved with 2x13 LFCC (13 Cepstrum coefficients and first time differentials), using 24 states and 96 symbols. In the matched situation, the best result on test data was achieved using a combination of all features extracted (47-dimensional), yielding 1% error rate or just below, at the optimal model size, 20-24 states and 48-54 symbols. These tests used a 1-second decision window size.

By using transcribed speech files it was also possible to compare the state assignment with the uttered phoneme for each segment. Investigating these results showed that the assigned states from the Viterbi search in a 3-state HMM, could be considered as phoneme classes

Introduction and background

Speech/Music discrimination (SMD) systems are used for different purposes. Before transcription in mixed audio database systems, a segmentation and classification pre-processor is often used to detect the speech segments. A sound classifier that could be an (SMD) system, is a natural component in such a pre-processor. Other applications could be music detection for listening purposes or to adapt appropriate signal processing on different kinds of sound, such as speech or music, in broadcasting. The design of an SMD system, including feature extraction, can be influenced by investigating the speech and/or the music signal. In this work, knowledge about the acoustic speech signal has influenced the design of the system.

Human speech can, in one aspect, be described as concatenated syllables. As a result, the acoustic signal varies between high energetic nuclei of the syllables surrounded by weaker segments. Even the spectral shape varies with the syllable rhythm. This behaviour is clearly recognized in a spectrogram, which is composed by quasi-stationary spectral states. Speech production could thus be considered as a state machine, where the states are phoneme classes.

Ajmera et al. (2003) have shown that the probability dynamism, which captures the dynamic behaviour of the probability values, is larger for a speech signal than for a music signal, while the entropy of a speech signal is smaller than for a music signal.

Similar observations in the speech signal have been reported, for example by Greenberg (1995). He describes a number of properties to be found in the speech signal from the perception perspective, for example the micro-modulation, in the interval of 3 to 12 ms, also considered as the pitch and the macro-modulation, 50-250 ms, associated with segments and syllables in the speech. Especially the macro-modulation supports this behaviour and it has been used in SMD tasks by the author in earlier reports (Karnebäck, 2001 and 2002). These reports showed that a feature based on the low frequency modulation, its amplitude and deviation (LFMAD), and especially a combination of LFMAD and MFCC's constitute robust features for SMD tasks.

Saunders (1996) used some of the characteristic features pointed out by Greenberg to successfully discriminate speech/music in FM broadcasting. Since the task was to develop a real-time system, he used only time domain features, mostly from ZCR (Zero Crossing Rate). Samouelian et al. (1998) also used some time domain features combined with two frequency features. Static spectral domain features which are used in speaker and speech recognition tasks can also be used for discriminating between speech, music and other sound sources. Hain & Woodland (1998) used MFCC (including normalised log energy and the first and second order time differentials) and Gaussian Mixture Models (GMM) separating four different sound sources (Speech, Music, Noise and Telephony speech). Gauvain et al. (1999) used a GMM system with MFCC as input features. They performed a segmentation and clustering step followed by a transcription session. Since MFCCs are often used in speech recognition systems for transcription, they were natural to use in the segmentation part too. Scheirer & Slaney (1997) pointed out and used some features for speech/music discrimination, which are closely related to the nature of human speech. Some of the features they used were the 4 Hz energy, percent of low energy frames and spectral centroid and spectral "flux" (Delta Spectrum Magnitude).

Nordqvist & Leijon (2002) used discrete HMMs to classify the acoustic environment in order to set gain and filter parameters in hearing aid equipment. By manually adjusting the constants for transitions they created a two-level system. They used 12 delta-cepstrum coefficients and modelled three sound sources (wide-band speech, traffic and telephony speech). The transitions between these sound sources were used by a second level HMM with four states. The objective was to decide whether the conversation was held on a telephone line or face-to-face.

Even though some features are already known to reflect this state switching in the speech signal, typically the energy feature, there might be other features with a strong impact from this behaviour. In order to get a tool to investigate this effect from individual features and feature combinations, a signal classification system is developed and presented in this report. The system is designed to discriminate between two signals and primarily evaluated on the SMD task. It could easily be further developed by arranging several discriminators in parallel or cascade for more complex classification tasks. The assumption in this work is that a music signal, albeit switching, does not show the same pattern as the speech signal and that it would be a good method to construct an SMD system using HMMs, since the dynamic behaviour is taken care of in the HMM. Other sound sources are also presumed to show different behaviour than speech on this matter.

HMMs are commonly used in speech recognition, speaker verification and speaker recognition systems. In SMD and classification tasks, HMMs are also used in different ways. HMM/ANN hybrid systems are used by Ajmera et al. (2003) and Williams & Ellis (1999). Continuous HMMs, containing one state per sound timbre and one model per sound source are used by Zhang & Kuo (1999). Allegro et al. (2001) also use per sound source individually trained HMMs. The most commonly used features are cepstrum coefficients, or secondary features derived from these (Ajmera et al., 2003; Williams & Ellis, 1999).

This report describes a sound classification system, evaluated on the SMD task, based on discrete ergodic HMMs, using both cepstrum coefficients and other features modelling speech and music sound sources separately. This is a first step in a feature evaluation work and the objective is to evaluate if the design is useful for its purpose by performing evaluations on some features and feature combinations. As a side-effect, preliminary evaluations of a phoneme classification test, investigating the quasi-stationary segments, could also be performed. The system is described in section 3 and the databases in section 4. In section 5, the technical data is presented together with a discussion on feature selection. The results are presented in section 6. In section 7, the result is discussed and conclusions are drawn.

Objectives and method

The objective of this work has been to design a sound classification system to be used mainly as an SMD system. The design is based on the assumption that the switching behaviour in the acoustic speech signal between phones or phone classes, is to a large degree speech specific and less found in other sound sources. The phone classes can be considered as quasi-stationary states and thus this assumption leads to the use of Hidden Markov Models (HMMs). Since HMMs are known as an effective method in automatic speech recognition tasks, it is a natural choice also for SMD tasks. The sound classification system should be used to evaluate the effect of the individual features on the discrimination or classification ability. The relation between the uttered phoneme and the states in the optimal path sequence should also be possible to investigate.

A system for training and testing was developed. The issues reported on in this paper are mainly

- feature selection
- number of observation symbols in the HMM
- number of states in the HMM
- feature evaluation
- decision window size
- matched and mismatched databases

The assumption that the states could be considered as phoneme classes was evaluated in the investigation of the assignment of each label. An argument for the use of discrete HMMs is that both the symbols and the states have some phonetic correspondence that can be evaluated. Transcriptions from the database used are available and comparison could be performed. By looking at each frame inside a label, a comparison between the assigned state, or symbol, and the uttered phoneme is performed. Only some of the feature combinations were examined on this issue.

System description

The proposed system is naturally divided in two parts, the test and the training subsystems, shown in Figs. 1 and 2 respectively. An overview is presented below and starts with the test system.

The test subsystem overview

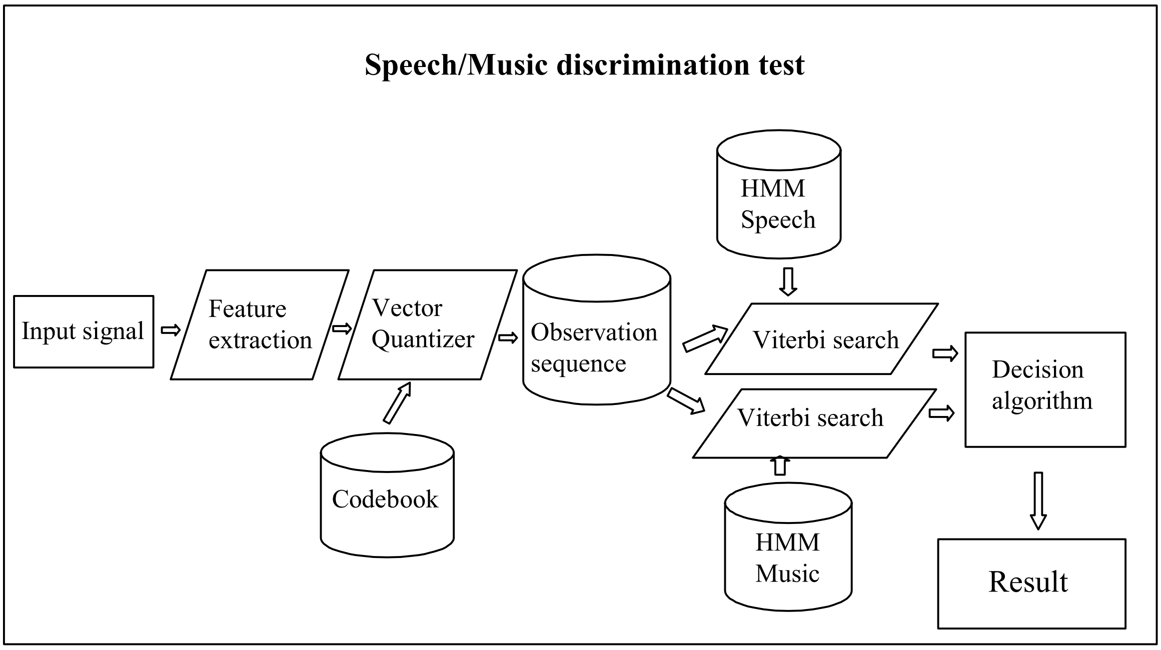


Figure 1. Test subsystem, see text.

The training system overview

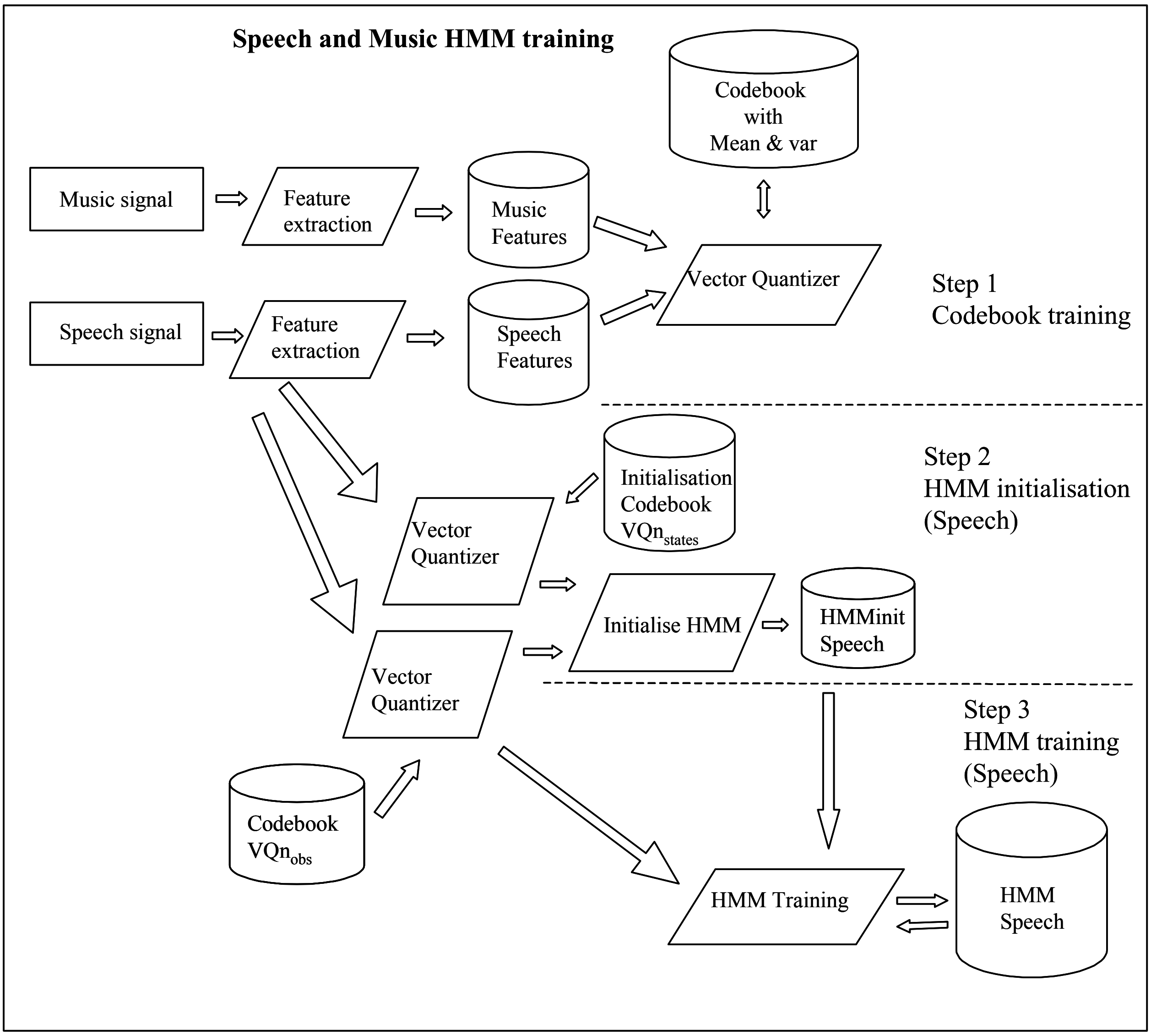


Figure 2. Training subsystem. VQnobs is the observation symbol codebook used for both training and testing. The codebook VQnstates is only used for training, see text. The Music model training is left out for steps 2 and 3, but it is performed in the same way as the speech model.

The system resembles a speaker verification system. The input signal is split into short-time analysis frames. From each frame is extracted a feature or feature combination. Features are normalised with mean and standard deviation derived from the training database, giving a distribution on the training database features of N(0,1). These values are stored together with the codebook. Only one codebook is used and shared by both the speech and music signals. The codebook is referred to as VQn, where n is the number of symbols or cells in the codebook. It would have been possible to build one separate codebook for speech signals and another for use on music signals, thus separating the two systems. Pinquier et al (2002b) uses separately trained models for speech and music respectively. The choice in this work was, however, to use common codebooks for the two systems. In the future, it would be possible to compare the performance for the two alternatives.

Observation symbols are obtained by a search in the codebook for each feature vector. A sequence of observations, corresponding to one or a few seconds, constitutes the decision window. The observation sequence is matched by a Viterbi search to the HMM speech and music models, respectively. One score value is calculated on each decision window at a certain frame rate.

From the Viterbi search a log probability was obtained that was used as a score in the decision algorithm. The decision rule for classification is

$$Class = \begin{Bmatrix} Music: & if \log P_{music} - \log P_{speech} > \eta \\ Speech: & if \log P_{music} - \log P_{speech} < \eta \end{Bmatrix}$$

(1)

where η is a threshold. This is equivalent to taking the log of the likelihood ratio and comparing with a threshold. No score normalisation was performed. The algorithm also gives the most probable path for the sequence. For a VQ3 assignment, it is possible to compare the observation sequence with the output state sequence from a three-state HMM.

The training system, Figure 2, is more complex than the test system. It can be divided into three parts: codebook training, HMM initialisation and HMM training.

Codebook training

Two codebooks are used. $\mathrm{VQ}_{\mathrm{nobs}}$ corresponds to the observation symbol set and $\mathrm{VQ}_{\mathrm{nstates}}$ to the states. The latter is only used to initialise the HMM. The codebooks were trained using the generalised Lloyd algorithm (Linde et al, 1980). The system is unsupervised, i.e. no labels are used to create the codebooks. The only manual work is to feed the system with split speech and music data files during the training phase.

HMM initialisation

The transition probability matrix **A** is initialised with large numbers (0.9) in the diagonal and the rest equally distributed.

$$\mathbf{A} = \{a_{ij}\} = \begin{Bmatrix} \dfrac{0.1}{(n_{states} - 1)} & if\ i \neq j \\ 0.9 & if\ i = j \end{Bmatrix} \qquad 2)$$

Since no element in the A-matrix is set to 0, the model is fully connected or ergodic. The observation probability matrix **B** is initialised using the quantified training data, from two separate codebooks. The matrix **B** is defined as

$$b_i(k) = P(x_t = o_k \mid q_t = s_i) \qquad (3)$$

where x_t is the observation at time t, o_k is symbol k from the observation symbol library $\mathbf{O} = \{o_1..o_{nobs}\}$, q_t is the state at time t and s_i is state number i. Since no states are defined, the codebook $\mathrm{VQn}_{\mathrm{states}}$ is used to initially assign each frame to a state. Let $V_{states}(1:T)$ be the vector of T consecutive quantified features from training data, using codebook $\mathrm{VQn}_{\mathrm{states}}$, $V_{obs}(1:T)$ is the training data quantified using the observation codebook $\mathrm{VQn}_{\mathrm{obs}}$ and T the number of frames in the training data. Equation (4) below is then used at initialisation.

$$b_i(k) = P(V_{obs}(t) = k \mid V_{states}(t) = i) = \frac{N_{obs_k, states_i}}{N_{states_i}}$$

(4)

where $N_{obs_k, states_i}$ is the number of joint events where $V_{obs}(t) = k$ and $V_{states}(t) = i$ and N_{states_i} is the number of single events where $V_{states}(t) = i$. No element in the B-matrix is initialized to zero. If the first initialization introduced zeroes they were replaced with small numbers. When the number of symbols was the same as the number of states in the HMM, the observation probability matrix was initialized in the same way as the A-matrix.

The Baum-Welch algorithm was used for training the HMMs.

Feature extraction

Several basic features were studied, listed in Table 1. They were combined in many ways, giving feature vector dimensions from 1 to 47.

The energy feature switches with the syllable rate. The nucleus of the syllable contains high energy and thus this feature was assumed to be useful. The binary energy feature only fortifies this effect.

Cepstrum parameters have been used in speaker verification with success and also in SMD systems (Hain and Woodland, 1998; Gauvain et al., 1999). Mel frequency scaled cepstrum coefficients, MFCC, are often used

Table 1. Basic features used in the investigation.

Feature	Description
Energy	Cepstrum coefficient 0, normalised, giving the log normalised energy.
High/low energy	Binary energy, from log normalised energy.
12 LFCC	Linear Frequency Cepstrum Coefficients (1-12) no energy, coefficients are normalised within each segment/file by mean cepstrum subtraction.
13 LFCC	Linear Frequency Cepstrum Coefficients (0-12), (including energy).
2 x 13 LFCC	Linear Frequency Cepstrum Coefficients (0-12), (including energy) and their first order time differentials. The differentials, delta values, were calculated on current frame ± 2 frames (giving 60 ms segment).
3x13 LFCC	Linear Frequency Cepstrum Coefficients (0-12), (including energy) and their first and second order time differentials. The differentials, delta values, were calculated on current frame ± 2 frames (giving 60 ms segment).
ZCR	Zero Crossing Rate in the frequency interval 0-8 kHz.
ZCR0-1	Zero Crossing Rate in the frequency interval 0-1 kHz.
ZCR1-6	Zero Crossing Rate in the frequency interval 1-6 kHz.
ZCR2-7	Zero Crossing Rate in the frequency interval 2-7 kHz.
Acf	Max value of an Autocorrelation function in the interval 4-20 ms, corresponding to 50-250 Hz.
Voiced/Unvoiced	Extracted from the Acf feature, using a threshold.
Spectral Gravity	Spectral centre of gravity in the frequency interval 0-8 kHz.

on speech signals. If they are appropriate for music signals is not yet known although Logan (2000) showed that MFCC outperforms LFCC, linear scaled frequency cepstrum coefficients, in short term discrimination tasks. The higher resolution in the lower frequency bands, achieved by MFCCs, reflects the auditory system and results in a higher yield on recognition tasks. This higher yield can also be obtained by using some more coefficients from LFCC. Since they do not differ in principle, but describe the same aspect in the signal, namely the overall shape of the spectrum, the choice in this work was to use LFCC.

The ZCR is supposed to detect the dominant frequency of the interval (Kedem, 1986), and is explored in several ways by Saunders (1996) and Scheirer & Slaney (1997). This feature obtains a high value at voiceless fricatives. When filtering in intervals, ZCR0-1, ZCR1-6 and ZCR2-7, the assumption was that the zero crossing rate should follow the dominant frequencies within the interval that could be a formant frequency in the speech signal.

Acf gives a high value for strong periodic signals and would typically obtain high values in voiced segments of the speech signal. A threshold in the Acf signal gives a binary feature, Voiced/Unvoiced. This is supposed to follow the voicing in the speech signal.

The spectral gravity reflects approximately the same character as the ZCR, only calculated in a different manner.

The features are selected from knowledge about the speech signal, assuming that the music signal, or any other sound signal, shows a different behaviour.

Special effort was dedicated to the question of feature selection, described in section 5.3.

Features were normalised with mean and standard deviation derived from the training database, giving a distribution on the training database features of N(0,1), before codebook training.

Databases

The speech and music databases were both divided into three parts, a training, a development and a test database. Model training was performed on the training database. The development database was used to find the threshold giving an equal error rate for speech and music data.

The database was composed in two different ways. One with a good match and one with mismatch between training and development on one hand and test data on the other hand. The matched database were composed in 5 different combinations, derived from the same sources. On many examinations, all five were examined and average values were calculated.

Two subsets of speech data with different sound quality were used, the Waxholm database (Bertenstam et al., 1995) and Swedish FM broadcasts. The Waxholm database is recorded in a silent room using a HiFi quality microphone with 16 kHz sample rate. Waxholm is a dialogue system where the task is to ask for boat schedule, hotel room and other facilities in the archipelago of Stockholm. It contains 68 speakers of mixed gender. The broadcasts were recorded using a standard FM receiver and a cassette tape recorder. 48 segments of speech with equal distribution between male and female speakers were collected during January and February 2001.

The music database also uses two subsets of different quality, CD recordings and FM broadcasts. The CD recordings contain a great variety of instrumental music and the broadcast recordings contain 46 segments of mixed kinds of instrumental music.

The speech databases differ more in quality than the music databases. The FM recordings contain more spontaneous speech than the short phrases in the Waxholm database. The SNRs differ approximately 12 dB on average and the transfer functions are different. The music databases are more similar, but the SNRs are higher on the CD recordings than in the radio segments. The purpose of these constructions was to evaluate the robustness of the system with a good and a bad match. The amount and mixture of data is presented in Table 2.

Initial silent parts were removed, but there is silence within the speech and at the end of the utterances, in both training and test data, while the music files were almost free from silence.

Experiments

Technical data

All signals were sampled at or resampled to 16 kHz with 16 bits in mono. The analysis window size was 20 ms and the frame rate was 100 Hz. The decision window was primarily one second. Before FFT-calculation a Hamming window was applied.

Evaluation methods

There are several parameters of interest to investigate in this system. As mentioned, different features and feature combinations are of interest and an initial investigation is presented in this report. However, the main focus is put on the impact of the model size on the SMD result. The database composition is also investigated. Results are presented as error rates in an SMD task.

Error rate calculations

The models were trained using the training databases. Evaluation was performed by calculating the equal error rate, on speech and

Table 2. Database compositions with approximate amount of data used from different sound sources. The matched database is recomposed in 5 different manners

Sound source	Mismatched database			Matched database		
Speech	Training	Development	Test	Training	Development	Test
Waxholm	15 min	8 min		9 min	5 min	9 min
FM-broadcasts			15 min	6 min	3 min	6 min
Total	15 min	8 min	15 min	15 min	8 min	15 min

Sound source	Mismatched database			Matched database		
Music	Training	Development	Test	Training	Development	Test
CD-recordings	15 min	8 min		9 min	5 min	9 min
FM-broadcasts			15 min	6 min	3 min	6 min
Total	15 min	8 min	15 min	15 min	8 min	15 min

music test data, when testing on the development databases, by varying the threshold, η, in Eq. 1.

This threshold is denoted η_{EER}. The speech error rate is calculated as

$$SER = \frac{N_{S_{false}}}{N_{S_{total}}} \cdot 100\% \qquad (5)$$

where $N_{S_{false}}$ is the number of falsely classified frames in the speech database and $N_{S_{total}}$ is the total number of frames in the speech database. The corresponding definition for MER is

$$MER = \frac{N_{M_{false}}}{N_{M_{total}}} \cdot 100\% \qquad (6)$$

where $N_{M_{false}}$ is the number of falsely classified frames in the music database and $N_{M_{total}}$ is the total number of frames in the music database. Index is used to inform what database is used in the test. For example SER$_{dev}$ gives the speech error rate from the development database. A total error rate (TER) was also calculated as the average of SER and MER.

$$TER = \frac{SER + MER}{2} \qquad (7)$$

Results are presented as EER$_{dev}$, which is the equal error rate on the development database (performed at $\eta = \eta_{EER}$), TER$_{test}$, which is the total error rate on the test database, also calculated at $\eta = \eta_{EER}$ and finally as EER$_{test}$, the equal error rate achieved on the test database, normally found at a threshold $\eta \neq \eta_{EER}$.

Feature selection

A method to help to select features with large impact on the discrimination ability was searched for. In many similar situations, PCA (Principal Component Analysis) is used in order to reduce the number of secondary features as input to the model-training algorithm. However, in this work the aim is also to evaluate the primary features and their impact on the discrimination ability. Fukunaga (1972) describes another possible way, also known as discriminant analysis. These methods should help to separate the sources in the static feature space. Since our system is based on the dynamic behaviour, modelled as HMM, and there is no known correlation between deviations in the static feature space and the dynamic behaviour, it is not likely that the methods would help. Besides the Vector Quantizer takes care of the separation in the static feature space.

As a tool in the feature selection procedure the correlation matrices were also investigated.

Results

The result presentation is divided into several parts, starting with some initial investigations concerning feature correlation and increasing number of observation symbols. The main results consist of error rates as a function of either number of states or number of symbols. These tests are performed on the two differently composed databases. Finally the decision window size is prolonged in one test.

Feature correlations

The results when investigating combined speech and music signals showed mainly weak correlations. Table 3 presents correlations above 0.4, while the others are left out for space-saving reasons.

Table 3. Feature correlation matrix. Both speech and music signals are included in the calculation. Only correlations larger than 0.4 (magnitude) are presented. First and second order differentials are not present in this investigation.

	Bin en.	Ceps-1	Ceps-2	Energy	ZCR	ZCR0-1	ZCR1-6	Acf
Energy	0.77							
ZCR		-0.55	-0.43					
ZCR1-6		-0.68	-0.46		0.51	-0.43		
ZCR2-7		-0.63				-0.40	0.66	
Acf	0.41	0.54		0.53				
Voiced/Unvoiced								0.49
Gravity		-0.57	-0.42		0.95		0.51	

Table 4. Feature correlation sub-matrix, speech signals only. Only features with any correlation larger than 0.4 (magnitude) are shown. First and second order differentials are not present in this investigation.

	Bin en.	Ceps-1	Ceps-2	Ceps-3	Energy	ZCR	ZCR0-1	ZCR1-6	ZCR2-7	Acf
Ceps-3		-0.40								
Energy	0.81			-0.44						
ZCR		-0.62	-0.46							
ZCR0-1	0.48	0.57			0.59					
ZCR1-6		-0.78	-0.55			0.61	-0.52			
ZCR2-7		-0.77		0.47		0.51	-0.41	0.67		
Acf	0.44	0.74			0.58		0.68	-0.59	-0.65	
Voiced/Unvoiced		0.41								0.59
Gravity		-0.62	-0.44			0.96		0.60	0.52	

Table 5. Feature correlation sub-matrix, music signals only. Only features with any correlation larger than 0.4 (magnitude) are shown. First and second order differentials are not present in this investigation.

	Bin en.	Energy	ZCR	ZCR0-1	ZCR1-6
Energy	0.75				
ZCR0-1			0.40		
ZCR2-7					0.60
Acf		0.45			
Gravity			0.94	0.49	

It was found that the correlations between features were much larger within speech features than within music features, see Tables 4 and 5. Similar observations, concerning differences in correlation between frequency bands, were also found when investigating the LFMAD feature described by the author (Karnebäck, 2001). The reason can be that the speech signal, when not disturbed, comes from one sound source while the music signal, when not solo, comes from several sources.

Note that in the music features no correlations between the cepstrum coefficients and ZCR, Acf and Gravity, above 0.4, were found. As expected, a large correlation was found between ZCR and Gravity in all signals.

The correlations indicate that the features from ZCR, Acf, Voiced/Unvoiced and Gravity would add some useful information to the conventional cepstrum coefficients, and improve the discrimination ability. The assumption is that the cepstrum coefficients would perform well in this task since they are used with good results in speaker recognition systems as mentioned earlier.

It was decided to perform the SMD tests on 13 LFCC (13-dimensional) as a baseline, and add either first (26-dimensional) or both first and second order time differentials (39-dimensional) of those, or ZCR, Acf and Gravity (16-dimensional). Tests were also performed on a combination of all the features presented (47-dimensional, referred to as 'All' feature) and all features except the first and second order time differentials (21-dimensional, referred to as 'All except delta'). A more thorough investigation of each feature candidate and more feature combinations has to be left for the future.

Initial investigations on number of observation symbols

Several speech/music discrimination (SMD) tests were initially performed on a separate small database changing primarily the number of observation symbols from 3 to 24. They showed that the error rate was dramatically reduced, from 11-15% down to 1-2%, with increasing number of symbols. This behaviour was found using three states in the HMM. Therefore it was decided to start with 24 observation symbols in the following tests.

State duration and phoneme classification

The average durations of the states were measured for two feature combinations on 3- and 4-states HMM, see table 6. Three and four states could correspond to some phoneme classes. The speech model was used on speech data and the music model was used on music data.

Table 6. Duration lengths on 3- and 4-states Hidden Markov Models. Durations are mean values of all states.

Feature	states	Mean duration (over all) Speech	Mean duration (over all) Music
13 LFCC	3	105 ms	190 ms
13 LFCC	4	99 ms	171 ms
All except delta	3	97 ms	121 ms
All except delta	4	84 ms	122 ms

The differences between speech and music durations are larger for the cepstrum features. The difference itself was expected and supports our assumption that the switching is more frequent in speech than in music signals. These results are also found and used for speech/music discrimination purposes by Pinquier (2002a).

Since the Waxholm database is transcribed, the classification per state could be compared with the transcription label showing the correspondence between the states produced by the Viterbi search and the labelled phoneme. Grouping all phonemes for each state indicates the assumed correspondence between state and phoneme class. In this procedure, every first and last frame in a transcribed phoneme was omitted to avoid transition problems. Hence, the system itself assigned each frame to a class, given by a number. The most frequent state assignment for each phoneme was supposed to correspond to that phoneme class. Considering this choice, an error rate was calculated for each class and in total. The total error on three classes, using 'All except delta', was approximately 12%. The error rate on 13 LFCC was higher, approximately 17%. Note, however, that the transcription is not strictly phonetic. The transcription was automatically performed, using the uttered text, literally, with manual corrections concerning deletions and insertions but not vowel changes due to assimilations. These results were found good enough and encourage further investigations.

SMD tests on mismatched databases

Effect of number of states

Experiments on the development and test databases from the mismatched databases were performed with three up to 48 states in the HMM models using 24 symbols on six feature combinations. Due to the mismatch in the database some features performed extremely well on $\mathrm{EER}_{\mathrm{dev}}$ but worse on $\mathrm{TER}_{\mathrm{test}}$ and $\mathrm{EER}_{\mathrm{test}}$. The models became over-trained. A randomised initialisation was used for the codebook training. As a result they did not become identical if tests were performed several times with new training procedure involved. Taking this fact into consideration the only conclusion to be drawn was a small reduction in error rate above 20 states on most of the feature combinations. Figure 3 shows the result for 24 states and 24 symbols.

The overall best result on $\mathrm{TER}_{\mathrm{test}}$, was achieved with 'All except delta', yielding 2.4% at 32 states ($\mathrm{EER}_{\mathrm{dev}}$ yielded 0%), although the differences between the explored features were small. Tests on the test database were performed on different thresholds and not only at $\eta = \eta_{EER}$. It was then found that the best test results could sometimes be achieved using a threshold close to the η_{EER} and sometimes far away. These results show the mismatch between the training and development databases on one hand and the test databases on the other hand, as expected. One explanation is that these materials are different, recorded under different circumstances giving different SNR etc. An illustration is found in Figure 5 below.

It is interesting to note that 3x13 LFCC does not show the best yield. It was also unexpected that 13 LFCC performs better than 3x13 LFCC up to 28 states. For tests on 28 states or more 3x13 LFCC performed equally well or better than 13 LFCC and 2x13 LFCC. The reason might be that the delta cepstrum force an undesirable splitting of the codebook. Other features are probably more important for this purpose, and if there were more symbols

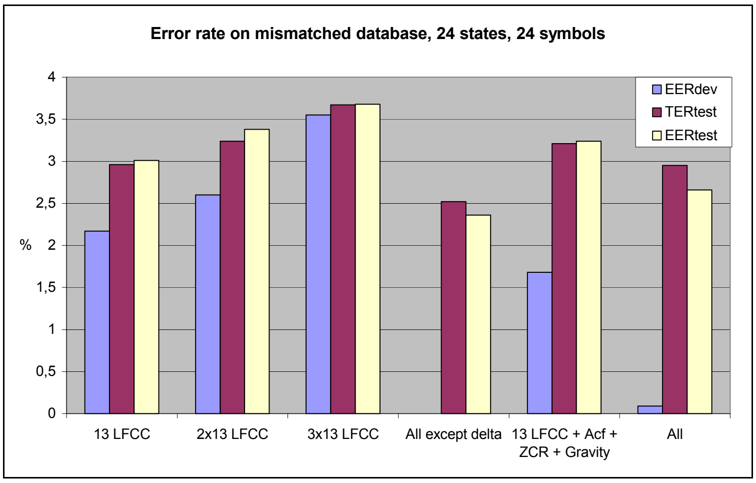


Figure 3. SMD test results presented as EER_{dev}, TER_{test} and EER_{test}, for some well performing feature combinations. The results show the mismatch between training and test databases.

in the codebook, the HMM would help to get a better result.

An inspection of one feature combination, 13 LFCC, showed that a large amount of the speech errors appeared in connection to a pause, speaker noise or breathing. To solve this problem a future system should be expanded with a silence detector together with the segmentation procedure.

Of great importance in a future SMD system is the robustness. The most common aspect of robustness is that there should be a small difference between the EER_{dev} and the TER_{test} result. However, a robust but poorly performing feature is not very useful. In general, the cepstrum parameters and especially 3x13 LFCC were found a little more robust on this aspect, as were 13 LFCC+Acf+ZCR+Gravity to some extent. The development database is half the size of the test database and thus the EER_{dev} is expected to be a little bit lower than TER_{test}. Since the error rates differed more than expected for all feature combinations, we cannot consider the system to be robust.

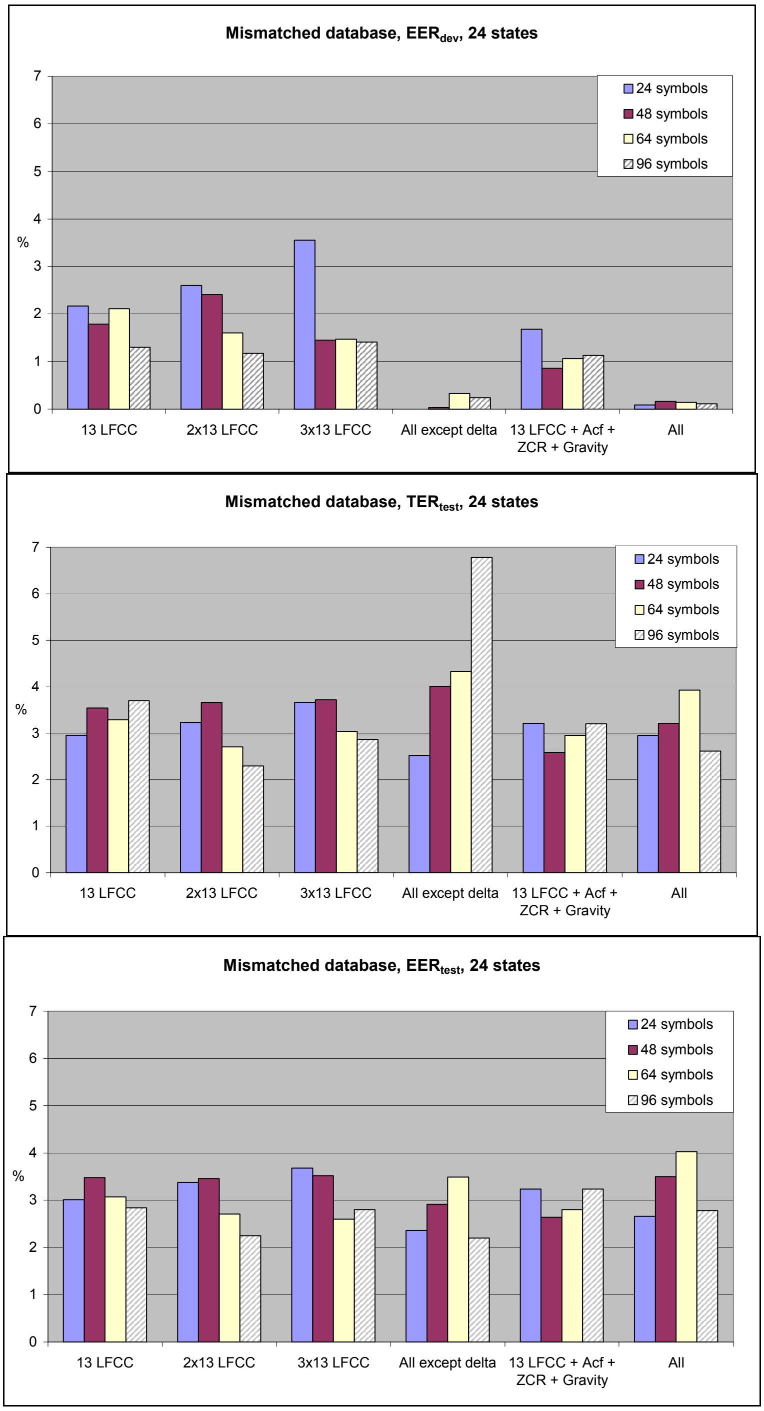
Effect of number of observation symbols

When investigating the effect of varying the number of symbols, no obvious conclusion could be drawn, see Figure 4. A tendency that increasing number of symbols (going from 24 to 96) would improve the results, especially for 3x13 LFCC, was however found. The problem with finding other conclusions was probably due to the mismatch in the database composition. 'All except delta' seemed to be a good feature when using 24 states and 24 symbols, but not when increasing the number of symbols.

The poor result on test data for ‘All except delta’ with 96 symbols is probably due to a very good match between training and EER data when increasing the codebook size, while the test database is different. It recognises the training data too well and therefore it cannot perform well on the test database. It's clearly over-trained.

The overall best test result using 24 states, was achieved on 2x13 LFFC with 96 symbols, yielding 2.3 % error on TER_{test}, while 'All except delta' yielded 2.2% on EER_{test}, using 96 symbols.

An alternative measurement of the robustness of the system and features is to calculate how many segments of music would have been falsely captured to detect at least 99% of the speech segments (the opposite could also be evaluated), see Table 7. In those tests, ‘All except delta' performed best with only 3.3% falsely captured music segments. This is a bit surprising considering the high error rate, 6.8%, but this rate comprises mainly speech errors and the music errors increase slowly when speech errors decrease on increasing threshold, see Fig. 5. When comparing the TER_{test} and EER_{test} results for 'All except delta', this effect is also clear. The TER_{test} is more depending on a good match between the databases than the EER_{test} is. Thus, for feature evaluation purpose, the EER_{test} is more relevant in a mismatched situation and the development database is thus of no use.



Mismatched database, EER_{dev}, 24 states

Mismatched database, TER_{test}, 24 states

Mismatched database, EER_{test}, 24 states

Figure 4. Results from SMD tests performed on 24 states as a function of number of observation symbols. EER_{dev} (top), TER_{test} (middle) and EER_{test} (bottom).

Table 7. Percent falsely detected music segments when detecting at least 99 % of the speech segments on test data using 24 states and 96 symbols in the HMM on mismatched databases.

Feature	13 LFCC	2x13 LFCC	3x13 LFCC	All except delta	13 LFCC + Acf + ZCR + Gravity	All
Percent falsely detected segments	9.5	10	12	3.3	27	20

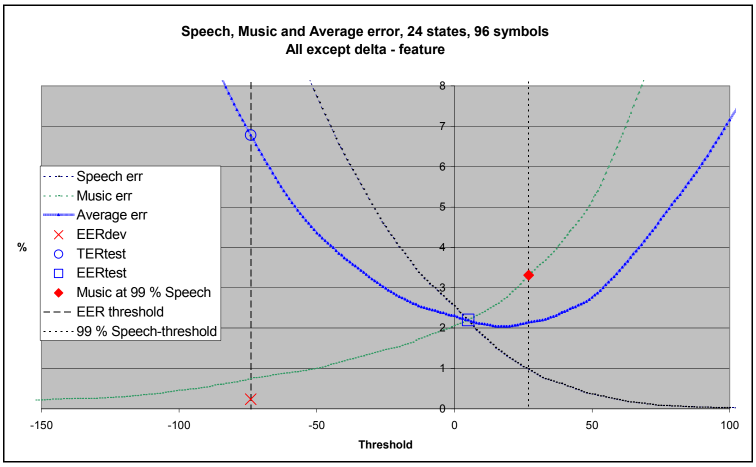


Figure 5. Speech, music and average error curves as a function of threshold value, using 'All except delta' with 24 states and 96 symbols. Note the unbalance between speech and music errors and that TER$_{test}$ (6.8 %) is much larger than EER$_{test}$ (2.2 %). EER$_{test}$ is however not the lowest average value. The point 99% speech detect and 3.3 % music error is marked in the graph, as well as EER$_{dev}$.

SMD tests on matched database

The system was not found to be robust against the mismatch in the databases. The EER$_{\text{dev}}$ was extremely good while test results were only fairly good. Thus, tests were continued on a matched database. Five pseudo randomly, recomposed database compositions were evaluated.

Baseline result with 24 states and 24 symbol

As a baseline and comparison with the mismatched database composition, tests were performed with 24 states and 24 symbols. Since these databases were very similar, the difference between EER$_{\text{dev}}$ and TER$_{\text{test}}$ became very small. As expected, TER$_{\text{test}}$ and EER$_{\text{test}}$ were also of the same size and much lower than in the mismatched case, see Fig. 6. When a matched situation occurs, TER$_{\text{test}}$ is more relevant and useful for system evaluation, while EER$_{\text{test}}$ was relevant for a mismatched situation, and only for feature evaluation purpose. The best result is now achieved with 'All' feature (1.3% on TER$_{\text{test}}$).

Effect of number of states

The effect of number of states were investigated on the matched database. The number of symbols were kept at 48, while the number of states were varied from 3 to 48. In Figure 7, four feature combinations are presented for one matched database composition. Due to processing time considerations, only one composition was tested with four feature combinations. The one chosen performed close to the average (slightly better) on the 'All' feature combination, which was evaluated on all 5 compositions, see Figure 8. It can be seen that ZCR, Acf and Gravity help to improve the result and that the 'All' feature combination performs best, 0.7 % for 32 states. A reduction in error rate between 12 and 32 states was found for 'All' and 'All except delta' but not seen at all for 13 LFCC and 3x13 LFCC.

A test to achieve a refined examination on number of states in the interval between 10 and 20 was performed on 13 LFCC using the same database composition. No surprise was found, maybe with one exception, a peak at 16 states on EER_{dev}. However, TER_{test} and EER_{test} did not vary much between 10 and 20 states. It should be noted that EER_{dev} in general varied more than TER_{test} and EER_{test}, and that 13 LFCC showed the largest variation among the features presented in Figure 7.

A small but clear error reduction is found, in Fig. 8, between 16 and 36 states. 20 or 24 states seem to be a good choice for best performance with 48 symbols, yielding 0.95 ±0.26 % and 1.01 ±0.36 % respectively on TER_{test}.

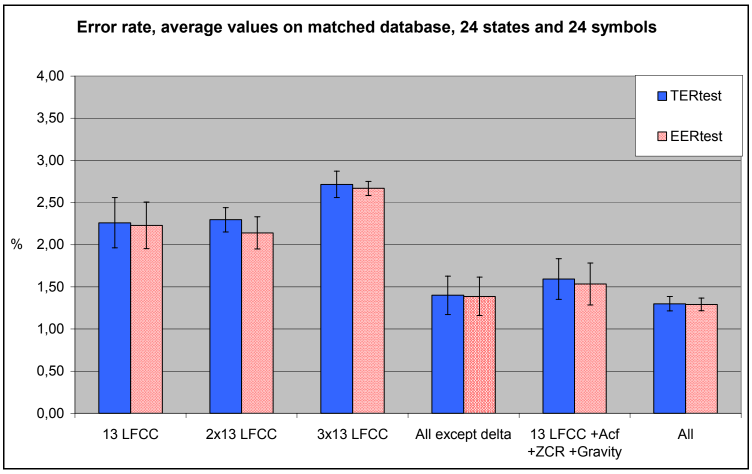


Figure 6. Average error rates from SMD tests performed on six feature combinations on five differently composed database compositions. 24 states and 24 symbols were used in the HMM. Results are presented together with ± 0.5 SD.

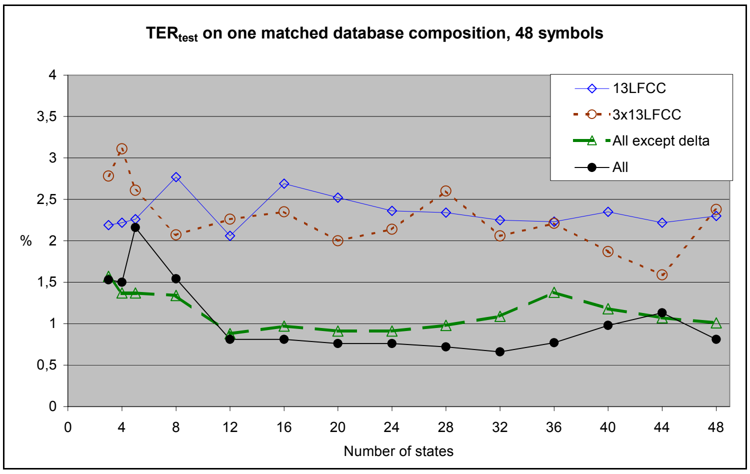


Figure 7. SMD test results as a function of number of states on four feature combinations on one matched database composition, see text.

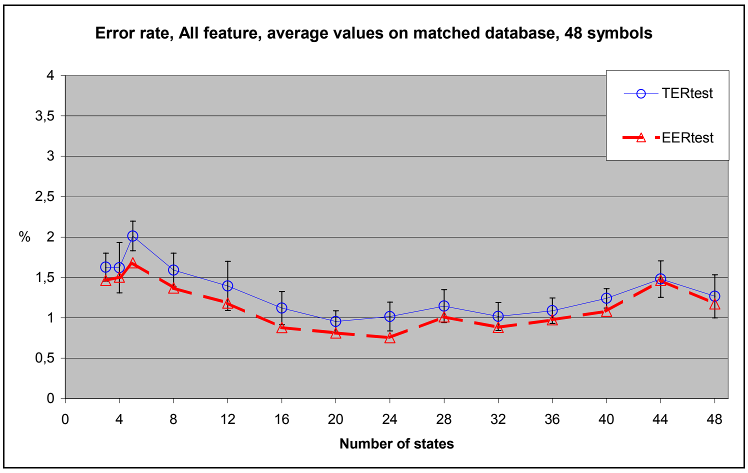


Figure 8. SMD test results as a function of number of states on matched databases for the 'All' feature combination. The average values for TER_{test} and EER_{test} from five database compositions are plotted. TER_{test} values are plotted together with ± 0.5 SD.

Effect of number of observation symbols

Also for the matched database, the effect of number of symbols were examined. Number of states were kept at 24. The first test comprises only 24 and 96 symbols for six feature combinations. Results are presented as average values for five database compositions in Figure 9. Since the difference between, EER_{dev}, TER_{test} and EER_{test} are fairly small in this test, only the TER_{test} results are presented. It can be seen that the features containing only LFCC (with or without the differentials) need more symbols to perform as well as those containing also ZCR, Acf and Gravity. The best result on 96 symbols was achieved with 13 LFCC + Acf + ZCR + Gravity (1.13%).

An investigation of falsely accepted music segments when searching for speech segments was performed. The result, presented in Table 8, shows less variation than for the mismatched situation. 13 LFCC + ZCR + Acf + Gravity performs best also in this aspect.

Figure 10 presents some error curves for this well matched situation, using the same database composition as in Figure 7, which should be compared with Figure 5. However, due to a better match, EER_{dev}, TER_{test} and EER_{test} are close to each other.

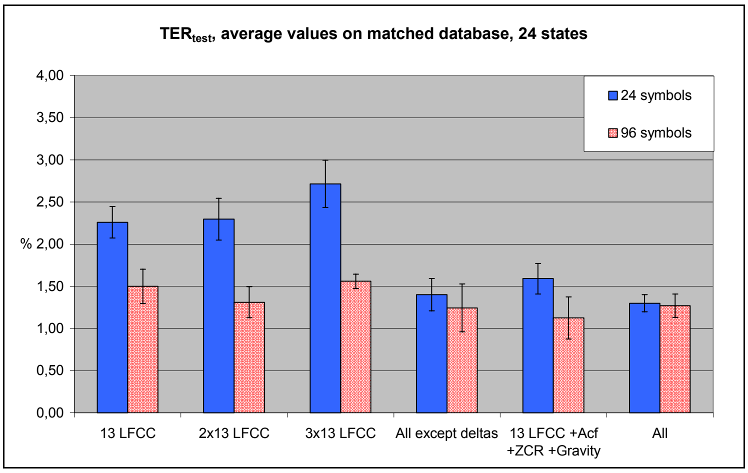


Figure 9. SMD test results for 24 and 96 symbols, 24 states, on matched databases. Average values on 5 differently composed databases. Results are presented with ± 0.5 SD.

Table 8. Percent falsely detected music segments when detecting 99% of the speech segments in test data on average for five database compositions. The HMM uses 24 states and 96 symbols.

Feature	13 LFCC	2x13 LFCC	3x13 LFCC	All except delta	13 LFCC + Acf + ZCR + Gravity	All
Percent falsely detected segments	1.8 ±0.8	1.4 ±0.4	2.3 ±0.7	1.4 ±0.7	1.2 ±0.6	1.4 ±0.5

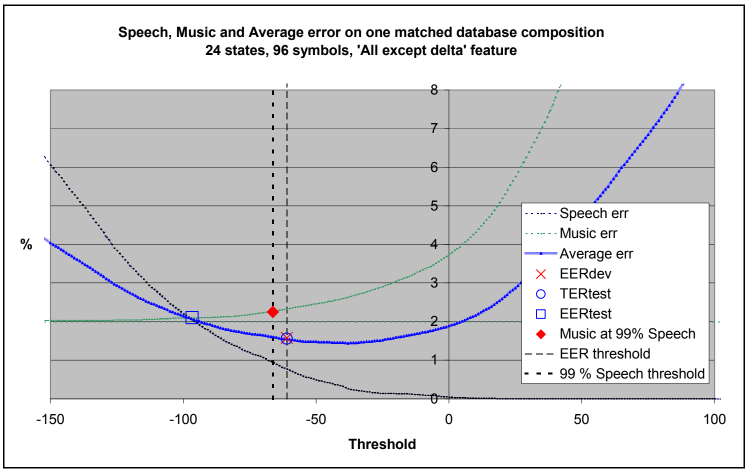


Figure 10. Speech, music and average error curves as a function of threshold value for one database composition, using 'All except delta' with 24 states and 96 symbols. Training, development and test data are well matched as can be seen. TER$_{test}$ (1.56 %) is equal to EER$_{dev}$. The point 99% speech detect and 2.25 % music error is marked in the graph, as well as EER$_{test}$ (2.10 %).

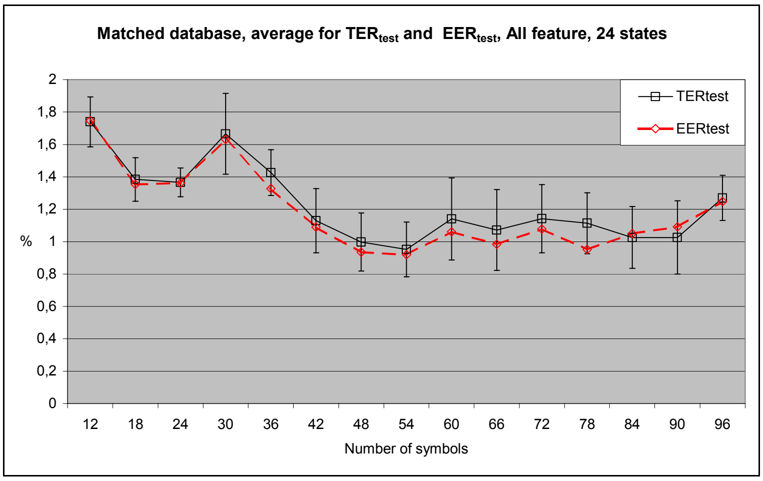


Figure 11. SMD test results as a function of number of symbols. 24 states were used in the HMM. Results are presented as average values for TER$_{test}$ and EER$_{test}$ from 5 database compositions. TER$_{test}$ values are plotted together with ± 0.5 SD

The second test on varying the number of symbols in finer steps was only performed with the 'All' feature combination. The number of states were kept at 24 while the number of symbols varied from 12 to 96 in steps of 6. A clear reduction in error rate is found at 48 (TER_{test}=1.0 ±0.36 %) and 54 (TER_{test}=0.95 ±0.34 %) symbols. The results for TER_{test} and EER_{test} are very similar, which can be seen in Figure 11.

Increased decision window size

When implementing an SMD system in an application, it will be working on different sizes of decision windows. Normally, a segmentation has been performed in an earlier stage, using criteria based on knowledge of the data to be retrieved and a cost function. Generally the performance increases with increasing window size.

In earlier reports (Saunders, 1996; Scheirer & Slaney, 1997; Williams & Ellis, 1999), a decision window size of 2.4-2.5 seconds were used, yielding 1.3 – 2% error rate. To get an indication on how good the result can be, tests were performed with a 2.4 seconds decision window size. Due to processing time considerations only one of the matched compositions were explored. Using 24 states with 24 symbols, the error rate was reduced by approximately 50% on average, compared with a 1-second decision window size. Several feature combinations achieved error rates below 1%, for example 'All except delta' achieved TER_{test} = 0.34% and 'All' feature TER_{test} = 0.55%. Using an optimal model size would improve these results further.

Feature evaluation

The feature evaluation has only started in this work and needs to be continued. Generally, it turns out that adding more features results in better performance, assuming that large enough models are trained. Adding Acf, ZCR and/or Gravity instead of the delta cepstrum features yields better result on smaller HMM models. The cepstrum and delta cepstrum features seem to be more general and robust, while ZCR, Acf and Gravity (seen as a cluster in this work) detect more specific cues in the signal, thus increasing the discrepancy between error rate on the development and test databases, when applied on a mismatched database. Acf probably captures the voiced speech segments, which are cleaner (higher SNR) in the training and development databases than in the test database (in the mismatched database composition). However, if the training and test databases are more similar, then these features perform very well. This can be observed in Figure 6 above where 'All except delta' yields 1.4% error rate (TER_{test} with 24 states and 24 symbols) which is better than the cepstrum features do by themselves. The best result, using 24 states and 24 symbols, was however achieved with the 'All' feature combination, yielding 1.3%.

Discussion and conclusion

A Speech/Music discrimination system using discrete Hidden Markov Models was designed. Several aspects of the system were investigated with focus on the HMM model size. Several different feature combinations were tested with models using up to 96 symbols and 48 states in the HMM. Different compositions of the database were tested, showing different behaviour for the features, on a good or a bad match between the training and test databases. The lowest error rate on test data, TER_{test}, with the mismatched composition was achieved with 2x13 LFCC using 24 states and 96 symbols, yielding 2.3%. When a good match occurs the best result was achieved with the 'All' feature combination, indicating that a good match helps to get use of all the features. The error rate was just below 1%, calculated on 1-second decision windows. The results must be considered as good and tests performed on 2.4 seconds indicated a 50% reduction of the error rate, approximately.

Even though the results in this work are in the same magnitude or slightly better than earlier reports on the same task (Saunders, 1996; Scheirer & Slaney, 1997; Williams & Ellis, 1999), they cannot be compared, since the databases are different. In this database, there was no singing within the music, for example. Earlier investigations (Karnebäck, 2002) show a 30% increase in error rate when including song in the music database. Other results reported, like Ajmera et al. (2003) do not use the same decision window size, thus making a comparison difficult.

Since these results show a much better yield than earlier tests on the same database (Karnebäck, 2001), which used only static models like GMM or VQ and MFCC features complemented with a low frequency modulation feature, LFMAD (some dynamic behaviour is, however, captured in the delta cepstrum and LFMAD features), the conclusion is that discrete ergodic HMMs, perform well in SMD tasks.

Generally, it seems that adding more features results in better performance, assuming that large enough models are trained, also found by Berenzweig & Ellis (2001). However, large feature dimensions need large models and a large amount of training data to get use of the large information embedded in the feature vector. When small models are desired, it could be useful to evaluate, in advance, what features to extract. In this work, it was found that adding Acf, ZCR and Gravity was a better choice than adding the delta cepstrum to 13 LFCC, on small models, but profound investigations are needed.

The second order delta cepstrum coefficients seem to either need very large models to increase the performance or they do not add much information at all. In this work, there was no test where 3x13 LFCC performed best. However, the ‘All' feature combination performed best, indicating that the Cepstrum features work well with other features and improve the result.

Some aspects of robustness were discussed. Which one to take into consideration depends on the application. When selecting speech segments for further transcription it is desirable to detect as many speech segments as possible with as few falsely detected music segments as possible. In this work, the 'All except delta' feature performed best on a mismatched situation, while several feature combinations performed almost equally well in the matched situation. The cepstrum features were less affected by the mismatched databases than feature combinations containing ZCR, Acf and Gravity.

The assumption that the phoneme classes are represented by quasi-stationary states in the HMM, could also be supported in this work. The agreement was approximately 12% and 17%, respectively, when an automatic phoneme classification for three and four classes was performed on the ‘All except delta' feature combination. These findings have to be further investigated on larger number of states. It is likely that a large degree of agreement will be found also on 20- or 24-state models, since the discrimination performance was found to be best in that interval. If so, the states can be considered as phonemes rather than phoneme classes.

Refined and extended experiments have to verify, or discard, the implication that 24 states and 48-54 symbols are optimal sizes of the models. Is this behaviour more enhanced for 'non-cepstrum' features? Can it be language specific? Another question to be answered is whether there are specific combinations of number of symbols and states in the HMM, that performs specifically well or bad.

The desired system for evaluating individual features or feature combinations in an SMD task, was designed. It was found useful for its purpose to investigate the impact from individual features on the SMD task. The error rates were found to be very small. Different ways to improve the performance on SMD tasks were discussed and they indicate that the system can be tuned to even better results. This tuning should be controlled by the specific application were the system should be a part. The system can also be used to investigate the agreement between the state assignment and the uttered phoneme on an individual feature basis.

Acknowledgement

This research was carried out at the Centre for Speech Technology, CTT, supported by Vinnova, KTH and participating Swedish companies and organisations.

References

Ajmera J, McCowan I & Bourlard H (2003). Speech/Music segmentation using entropy and dynamism features in a HMM classification framework. *Speech Communication* 40: 351-363.

Allegro S, Büchler M & Launer S (2001). Automatic Sound Classification Inspired by Auditory Scene Analysis. *Workshop, Eurospeech.*

Berenzweig A & Ellis D (2001). Locating singing voice segments within music signal. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Bertenstam J, Blomberg M, Carlson R, Elenius K, Granström B, Gustafson J, Hunnicutt S, Högberg J, Lindell R, Neovius L, de Serpa-Leitao A, Nord L & Ström N (1995). The Waxholm Application Data-Base. *Proc of Eurospeech*, 1: 833-836.

Fukunaga K (1972). *Introduction to statistical pattern recognition*. Academic Press Inc. ISBN 0-12-269850-9.

Gauvain JL, Lamel L, Adda G, Jardino M (1999). recent advances in transcribing television and radio broadcasts. *Proc of Eurospeech*, 2: 655-658.

Greenberg S (1995). The ears have it: The auditory basis of speech perceptions. *International Congress of Phonetic Sciences*, 3: 34-41.

Hain T & Woodland P (1998). Segmentation and classification of broadcast news audio. *Intl Conf Spoken Language Process*, 6: 2727-2730.

Karnebäck S (2001). Discrimination between speech and music based on a low frequency modulation feature. *Proc of Eurospeech*, 1891-1894.

Karnebäck S (2002). Expanded examinations of a low frequency modulation feature for speech/ music discrimination*.* *Intl Conf Spoken Language Process*., 3: 2009-2012.

Kedem B (1986). Spectral analysis and discrimination by zero-crossings. *IEEE, Proceedings*, 74/11: 1477-1493.

Linde Y, Buzo A & Gray RM (1980). An algorithm for vector quantizer design. *IEEE Trans Commun COM-28*: 84-95.

Logan B (2000). Mel frequency cepstral coefficients for music modeling. *Intl Symp on Music Inform Retrieval.*

Nordqvist P & Leijon A (2002). Automatic classification of the telephone listening environment in a hearing aid. *TMH-QPSR, KTH*, 43/2002.

Pinquier J, Rouas J-L & André-Obrecht R (2002a). Robust speech/music classification in audio documents. *Intl Conf Spoken Language Process*, 3: 2005-2008.

Pinquier J, Sénac C & André-Obrecht R (2002b). Speech and music classification in audio documents. *Intl Conf Acoustics, Speech & Signal Proc.*

Samouelian A, Robert-Ribes J & Plumpe M (1998). Speech, silence, music and noise classification of tv broadcast material, *Intl Conf Spoken Languange Proc,* 3: 1099-1102.

Saunders J (1996). Real-time discrimination of broadcast speech/music. *Intl Conf Acoustics, Speech & Signal Proc*, 2: 993-996.

Scheirer E & Slaney M (1997). Construction and evaluation of a robust multifeature speech/music discriminator. *Intl Conf Acoustics, Speech & Signal Proc*, 2: 1331-1334.

Williams G & Ellis D (1999). Speech/music discrimination based on posterior probability features, *Proc of Eurospeech*, :2: 687-690.

Zhang T & Kuo J (1999). Hierarchical classification of audio data for archiving and retrieving, *Intl Conf Acoustics, Speech & Signal Proc*, 6: 3001-3004.