



Research Center

NRC-TR-2008-005

On Personal Content Management Techniques for Audio

Tommi Lahti¹, Marko Helén², Olli Vuorinen³, Eero Väyrynen⁴,
Juha Partala⁴, Johannes Peltola³, Satu-Marja Mäkelä³

Nokia Research Center, Personal Content & Media Team, Tampere, Finland¹,
Tampere University of Technology, Department of Signal Processing, Tampere, Finland²,
VTT Technical Research Center of Finland, Oulu, Finland³,
University of Oulu, Oulu, Finland⁴

June, 20, 2008

Abstract:

State-of-art automatic analysis tools for personal audio content management are discussed in this paper. Bayesian networks based audio classification algorithm provides classification into four main audio classes and serves as a first step for other subsequent analysis tools. For speech analysis we propose an improved BIC based speaker segmentation and clustering algorithm and a combined gender and emotion detection algorithm utilizing prosodic features. For the other main classes it is often hard to devise any general and well functional pre-categorization that would fit the unforeseeable types of user recorded data. For compensating the absence of analysis tools for these classes we propose the use of efficient audio similarity measure and query-by-example algorithm with database clustering capabilities. Based on the experiments the audio similarity framework is also capable of producing relationship metadata for example relating the labeled speaker segments of one sample across the whole user's personal database.

By following some simple combined implementation principles the framework can be supported also in personal mobile devices. The experimental results show that the combined use of the algorithms is feasible in practice.

Index Terms:

Personal audio content management
Speaker segmentation and clustering
Gender and emotion detection
Query-by-Example
Audio similarity and classification

ON PERSONAL CONTENT MANAGEMENT TECHNIQUES FOR AUDIO

Tommi Lahti¹, Marko Helén², Olli Vuorinen³, Eero Väyrynen⁴,
Juha Partala⁴, Johannes Peltola³, Satu-Marja Mäkelä³

Nokia Research Center, Personal Content & Media Team, Tampere, Finland¹,
Tampere University of Technology, Department of Signal Processing, Tampere, Finland²,
VTT Technical Research Center of Finland, Oulu, Finland³,
University of Oulu, Oulu, Finland⁴

ABSTRACT

State-of-art automatic analysis tools for personal audio content management are discussed in this paper. Bayesian networks based audio classification algorithm provides classification into four main audio classes and serves as a first step for other subsequent analysis tools. For speech analysis we propose an improved BIC based speaker segmentation and clustering algorithm and a combined gender and emotion detection algorithm utilizing prosodic features. For the other main classes it is often hard to device any general and well functional pre-categorization that would fit the unforeseeable types of user recorded data. For compensating the absence of analysis tools for these classes we propose the use of efficient audio similarity measure and query-by-example algorithm with database clustering capabilities. Based on the experiments the audio similarity framework is also capable of producing relationship metadata for example relating the labeled speaker segments of one sample across the whole user's personal database.

By following some simple combined implementation principles the framework can be supported also in personal mobile devices. The experimental results show that the combined use of the algorithms is feasible in practice.

Index Terms— Audio classification, personal audio content management, speaker segmentation, speaker clustering, gender detection, emotion detection, query-by-example, audio similarity.

1. INTRODUCTION

Until recently, the content analysis tools have been mostly directed for professional use. This has not been a big surprise because of the huge amount of professionally produced audio data and the lack of digital personal recordings. Now, with the multimedia mobile devices that are almost always close at hand the situation has been rapidly changing. Just think what kind of explosion with personal videos has happened in the Internet. Heavily increasing amounts of self-created data makes searching, organizing, and sharing challenging if everything is to be done manually by the user. In this respect, various search applications and also more innovative smart applications can readily benefit from automatic content management and metadata creation.

By audio classification it is often meant the classification (or segmentation) of a general audio sample into a number of representative audio classes. The basic classification set required in audio/video content analysis includes the classes silence, speech, music, and environmental sounds [1]. Subsequently for

example speaker segmentation and clustering [2], [3], [4], [5], as well as gender and speaker emotion detection [6] can be performed on speech category data.

From the personal audio content management point of view the situation with the subsequent analysis tools for the other main audio classes is not that well-defined. The fundamental problem is that it is hard to device any general and well functional categorization in advance that would fit the type of audio data the user is likely to record. Running many parallel single purpose metadata extractors would be computationally expensive and unfeasible solution especially in portable devices.

Clustering based on audio similarity does not assume any pre-categorization and provides an implicit way of organizing unforeseeable user data automatically. Efficient similarity measures for personal audio content management purposes have been recently proposed in [7], [8], and [9].

This paper is about centralized metadata creation framework for personally created audio data and is organized as follows. Bayesian network based audio classification algorithm is discussed in Section 2. Section 3 considers speech analysis tools for speaker segmentation and clustering and the gender and emotion detection algorithm is covered in Section 4. In Sections 5 and 6 the novel similarity measure and the query-by-example algorithm is discussed. Section 7 summarizes the experimental evaluations. Finally, in Section 8 conclusions are made.

2. AUDIO CLASSIFICATION ALGORITHM

The general audio classification algorithm is used as a preprocessing step for further analysis. In our approach the most important class to detect is speech. Speaker Change Detection (SCD), GENDER Detection (GEND), and EMOTION Detection (EMOD) tools all rely on correct speech detection results. We have slightly improved our earlier implementation [10] by incorporating fluctuation pattern features [11] which help in differentiating some of the problematic cases we had earlier especially with speech and music. To cope with the rest of the classes an efficient SIMilarity measure and query-by-example (SIMqbe) algorithm is utilized. With the algorithm, high performance implicit grouping and refined modeling for otherwise unseen data cluster is obtained and not all the classification duties are left to be handled by the audio classification algorithm.

A simple Bayesian network was selected as the topology of the classifier. Bayesian networks are directed graphs which model joint probability distributions, and are a classic choice in statistical classification schemes [12].

The Bayesian classifier and selected audio classes of the network have previously been shown to be capable of robust audio classification [10], [13]. They also showed good performance even for challenging mobile material.

Our network structure consists of N binary decision nodes D_1, D_2, \dots, D_N that are arranged hierarchically to separate the audio classes in a sequence into $N+1$ discrete classes C_1, C_2, \dots, C_{N+1} (see Figure 1). A set of audio features $F_i \in F = \{X_1, X_2, \dots, X_K\}$ is provided for each node D_i as an input so that different nodes may be connected to different set of audio features. Each node is associated with the class and non-class feature probability distribution models $M_{C_i}(F_i; \lambda_{C_i})$ and $M_{\bar{C}_i}(F_i; \lambda_{\bar{C}_i})$, respectively, where the semicolon notation is used to emphasize the trained parameters of the model. Each node also takes the non-class probability as an input from the previous node, if there is one. In case $i=1$, it is agreed that $M_{\bar{C}_0}(F_0)=1$. In our case the class and non-class feature probability distributions are modeled by using Gaussian distribution modeling. The training material for the class feature distributions contains the corresponding audio samples for the given class and the remaining samples that correspond to the classes below the current node in hierarchical network are used for training the non-class model. The joint probability density function for the network and hence the network class probabilities is given by the formula

$$P(C_i|F_i) = M_{C_i}(F_i; \lambda_{C_i}) \cdot M_{\bar{C}_{i-1}}(F_{i-1}; \lambda_{\bar{C}_{i-1}}), \quad (1)$$

where $M_{\bar{C}_0}(F_0)=1$.

The implemented network consists of four binary decision nodes that are arranged hierarchically to separate the audio classes in a sequence into five elementary classes. Each node models the class/non-class decision, which uses Gaussian distribution modeling of the selected audio features in each decision node. The class probabilities in our network are not conditionally dependent to each other, thus the sum of all $P(C_i|F_i)$ is not one and the results need to be normalized.

The network is structured as shown in Figure 1. The bottom row represents the basic acoustic features calculated from the signal and which are selected to describe the each audio class/non-class. The darker nodes represent the Bayesian decision nodes, the arrows denoting statistical dependencies. Each decision node is connected to a set of features that are described in more detail in Table 12 at the end of the paper.

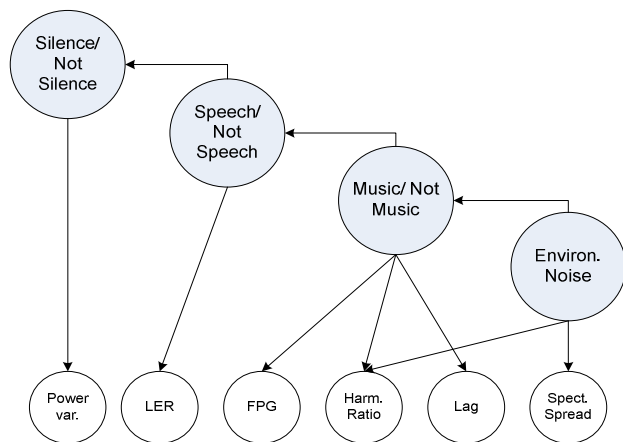


Figure 1: The Bayesian network used for audio analysis.

First hierarchy level calculates the probability for silence class, second for speech, third for music, and the last node evaluates the probabilities for constant and inconstant environmental sounds. This approach handles the “broad” environmental sounds class more reliably and resulted in improved performance in our experiments. In the end the two noise classes are combined into a single environmental sounds class. Classification is performed separately for each 3 second segment and inside the segment required audio features are calculated in shorter analysis windows. The feature values are then averaged. Classification results achieved for each 3 s segment are post filtered by averaging the results of 3 consecutive segments. Each audio segment is assigned with the audio class probability from all the decision nodes. The class that receives highest probability is the result of the audio classification.

The classification topology allows the advantage of a hierarchical classifier, while the conditional dependence between the non-class probabilities from previous nodes and the probabilities of the current node helps to cope with errors made in the early stages of the hierarchy. By keeping the structure simple and using a well-known basis for the classifier we are able to maintain robustness in the presence of noise and low-quality input material.

3. SPEAKER SEGMENTATION AND CLUSTERING

The goal of the speaker segmentation and clustering is to find the boundaries for speaker segments and detect which segments are spoken by the same speaker. Typically speech from the same speaker may appear multiple times in an audio stream. In mobile device applications speaker metadata available from speaker clustering is useful for indexing and browsing of audio and video data. Clustering can also be used to accumulate longer speech segments for subsequent processing e.g. speaker adaptation, speaker identification, speaker emotion detection etc.

A block diagram of the speaker segmentation system is shown in Figure 2.

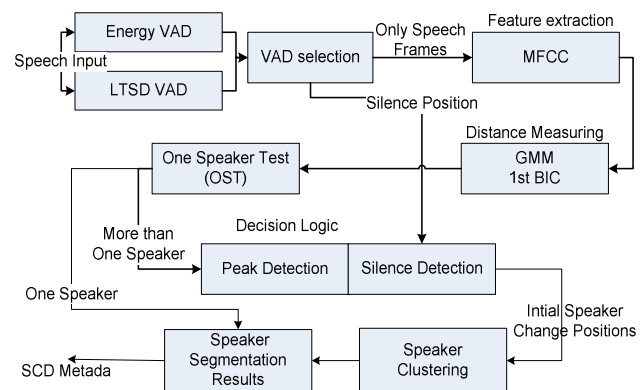


Figure 2: Block diagram of speaker segmentation system.

In the first step input frames are classified into speech or silence category. In our approach this step is executed by using combined Voice Activity Detector (VAD), which is relying on two VAD implementations. Combined VAD utilizes conventional energy based VAD and Long Term Spectral Divergence (LTSD) VAD which is known to be especially noise robust VAD implementation [14]. The VAD that finds more silence frames is used in combination with peak detection to indicate the initial speaker change positions.

For speech frames Bayesian information criteria (BIC) based dissimilarity measurement is performed. This phase of the system is discussed more in detail in Section 3.1.

One speaker test (OST) is done to detect if the recording contains speech only from one speaker. The test is based on the BIC-ratio test which is described below.

After the initial speaker changes are detected they are used as input for speaker clustering algorithm, which clusters segments and gives them a proper speaker label. If two adjacent segments belong to the same speaker, they are merged. Finally, segmentation metadata is extracted. More detailed description of the used Speaker Change Detection (SCD) system, however, without the presence of the proposed speaker clustering phase is presented in [3].

3.1 Bayesian Information Criteria

BIC, being today one of the most commonly used methods for SCD, was first proposed by Chen & Gopalakrishnan [2]. The BIC is a maximum likelihood criterion penalized by the complexity of model parameters. A one data segment has two hypotheses. It either contains speech from one speaker when there exists a single Gaussian model or it contains speech from two speakers with two multidimensional Gaussian models. The maximum likelihood ratio between the two hypotheses is then formulated as

$$R(i) = \frac{N_x}{2} \log |\Sigma_x| - \frac{N_{x1}}{2} \log |\Sigma_{x1}| - \frac{N_{x2}}{2} \log |\Sigma_{x2}|, \quad (2)$$

where Σ is the corresponding covariance matrix, N is the number of acoustic vectors in the complete sequence and x corresponds to the combined data segment of segments $x1$ and $x2$. The variations between one speaker (one Gaussian) and two speakers (two different Gaussians) is given by

$$\Delta BIC(i) = -R(i) + \lambda P, \quad (3)$$

where P is the penalty term $P = \frac{1}{2} (p + \frac{1}{2}p(p+1)) \times \log N_x$ and p is the dimension of the acoustic space and λ is the penalty factor. The negative value of BIC denotes the speaker turn change in the sequence.

The BIC is achieved by comparing Gaussian distributions G_{x1} and G_{x2} calculated for two adjacent windows to Gaussian distribution G_x calculated for window including both smaller windows as illustrated in Figure 3.

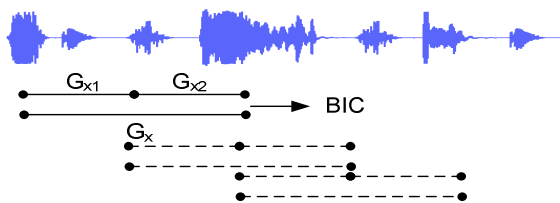


Figure 3: Example of window sliding for BIC.

3.2. Speaker Clustering

Typically BIC distance measure is applied to adjacent speech segments as in [5] for detecting or evaluating speaker change points. Problems in this approach include difficulties of setting proper thresholds and dealing with short data segments. Our approach is different and is based on the assumption that, if sequences belong to the same speaker, their BIC distance relations, which we call BIC profiles, to all other segments are

mostly similar [3]. BIC distance matrix is calculated between all detected speech segments. Segments are composed based on initial speaker change positions from SCD, see Figure 2. BIC matrix can be presented as:

$$BIC_{Matrix} = \begin{bmatrix} BIC(S_{1,1}) & BIC(S_{1,2}) & \dots & BIC(S_{1,i-1}) & BIC(S_{1,i}) \\ BIC(S_{2,1}) & BIC(S_{2,2}) & \dots & & BIC(S_{2,d}) \\ \vdots & & \ddots & & \vdots \\ BIC(S_{j-1,1}) & & & & BIC(S_{j-1,i}) \\ BIC(S_{j,1}) & BIC(S_{j,2}) & \dots & BIC(S_{j,i-1}) & BIC(S_{j,i}) \end{bmatrix}, \quad (4)$$

where $BIC(S_{i,j})$ is a BIC value calculated between speech segments initially labelled as i and j . Segment indexes i and j get values from one to the number of segments.

In Figure 4 are illustrated BIC profiles, including five speech segments from three different speakers. One speech segment is from speaker1, two segments from speaker2 and two segments are from speaker3. Speaker labels corresponding to segment index in Figure 4 are: 1, 2, 3, 2, 3.

The proposed clustering algorithm uses the BIC profile information for creating the clusters. The determination of the cluster is simply done by selecting one BIC profile to represent each speaker. We call this selected BIC profile here as Representative Speaker Cluster (RSC) profile. To measure the closeness between the already selected RSC profile and the candidate profile, simple residual-mean-square -difference is used. It is calculated by subtracting candidate profile from RSC profile and calculating variance from the residual values [3].

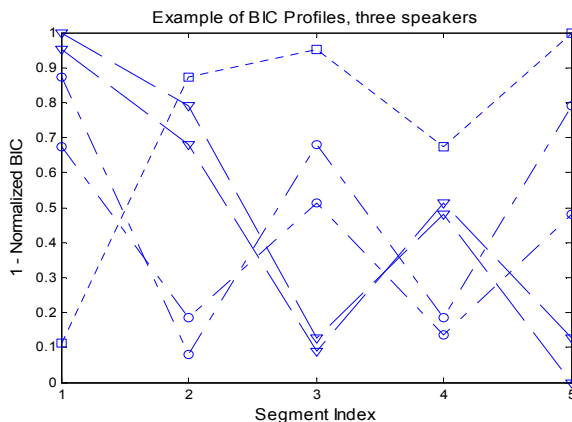


Figure 4: Example of BIC profiles. (square=speaker1, circle=speaker2, and triangle=speaker3).

Clustering of audio segments is performed divisively from top to down. Comparing with the basic hierarchical clustering algorithms, the additional part is the BIC -ratio test in each node used to test whether the data is originally from one speaker only. BIC-ratio is calculated by dividing the minimum value of the BIC matrix by the maximum value of the BIC matrix [3]. This estimates the biggest variation between segments in a current situation. The maximum value is one, which indicates that compared segments are homogenous. If the ratio is above the experimentally set threshold (0.5) it is understood that all the segments actually corresponds to one speaker only.

Steps of Speaker Clustering:

- I. Initialization:
 - Calculate the BIC matrix.
 - Calculate the BIC profiles.
 - Calculate the BIC-ratio.
- II. If BIC-ratio > threshold

Stop splitting the cluster.

III. *Candidate profile selection:*

Find two profiles, which have biggest difference.

Difference can be calculated from BIC-matrix or using RMS-difference between profiles.

IV. *Calculate the RMS-distance between the two candidates*

If RMS-distance < threshold

Stop clustering the cluster.

Else

Accept both candidate profiles as proper RSC-profiles.

V. *Create two clusters out of one by clustering all remaining BIC-profiles in the original cluster to the closest RCS-profile according to the RMS-distance.*

VI. *Repeat the process for each resulted sub-cluster.*

When all representative profiles are found, segments are labeled with the same label as the closest RSC profile. More detailed description of the Speaker Clustering algorithm has been presented in [15].

4. EMOTION AND GENDER CLASSIFICATION ALGORITHM

An acoustic analysis was performed on the speech data in order to classify the emotional content of each speech sample to four basic emotion classes: neutral, happy, sad, and angry [6]. The vocal parameters of basic emotions are now understood relatively well [16], and effective algorithms have been developed to estimate them from data. The following algorithm was implemented for estimating various prosodic features from speech and classifying them into several emotion classes [17]. In our framework robust gender detection was performed with the same algorithm in order to take full advantage of prosodic feature computations and the algorithm implementation.

4.1. V/UV segmentation

A digital audio signal is first partitioned into 60ms overlapping segments in 10ms steps. Then the cepstrum is computed for each segment, and voiced/unvoiced (V/UV) classification is performed by combining information from consecutive segments. Cepstrum peaks are estimated in two stages. First, rough estimates are computed, and then more accurate values are achieved.

For computing rough estimates, a cepstrum is computed for each segment. An amplitude correction by linear weighing is performed on the cepstrum in range of 1/700 – 1/40 quefrenencies in order to compensate for fundamental frequency (F0) variation within a segment. This operation enables F0-independent global thresholding for peak detection. It also enables better F0-estimation at the end points of voiced segments. To emphasize the F0-peaks of a noisy cepstrum and thus making peak detection more reliable, a running average liftering over the cepstrum is performed. Medians of the cepstrum peak amplitudes and segment root-mean square energies over the speech recording are calculated next. These are used in the second stage as thresholds to find the cepstral peak locations of voiced segments. If multiple peaks are present within a segment the one lowest in quefrenency is selected. The peak detection operation is embedded in a F0-tracking routine that uses a 2 msec tolerance window for locating the next expected pulse peak in the signal segment. This function is designed to enhance the processing of trailing voiced segments.

A common problem that makes F0-estimation difficult is the frequency doubling caused by higher formants of speech.

This problem was solved by applying a nonlinear function to the signal amplitude prior to cepstrum calculations as was previously done in [17]. By flattening the spectrum it reduces the predominance of higher formants. This algorithm also improves glottal pulse peak determination of creaky voiced segments by stabilizing the amplitude of consecutive cycles.

Finally, a segment is classified voiced if it and the segment immediately before it have root-mean square energy and cepstral peak amplitude higher than the corresponding median-based thresholds. Consecutive voiced segments and segments with only one unvoiced segment between them are then joined to form the final V/UV segmentation data.

4.2 F0-contour estimation

A waveform-matching algorithm is used to estimate the F0 pitch contours for each voiced segment. Accurate F0-estimation is required in order to estimate perturbation features such as jitter and shimmer. First, a Finite Impulse Response (FIR) band-pass filter is selected from a filter bank according to the F0 distribution. The distribution is obtained from the rough pitch information during V/UV segmentation from the cycle period information of cepstrum peak locations. A zero-crossing calculation is then used to construct rough cycle boundaries for the waveform-matching algorithm. Every consecutive pair of roughly marked cycles of 55Hz FIR high-pass pre-filtered data is then screened for maximum or minimum peak match using least squared error method with quadratic peak interpolation. The raw cycle peak frequency contour is then screened for simple errors and fitted with a linear smooth contour for prosodic parameter calculations.

4.3 Feature selection and statistical classification

From the V/UV segmentation and F0 contour data initial set of 46 prosodic features were calculated automatically to cover both emotion and gender detection needs. The selected features and the selection process are summarized in Section 7 describing the experimental setups. Classification is performed using the k-Nearest-Neighbour (kNN) classifier. In a training phase a set of prototypical feature vectors from each class are stored in the classifier memory. An unknown feature vector is then compared to all prototypes, and k closest (in vector space) vectors are picked up. A majority voting is performed among these to identify the class in which most prototypes belong. The unknown feature vector is decided to belong to that class.

5. AUDIO SIMILARITY ESTIMATION

For similarity estimation, basically any distance measure can be used. In previous tests by Virtanen and Helén [9], the Euclidean distance between probability density functions (pdfs) of feature distributions provided the best results and thus, it is used here. The similarity between two samples is estimated by the square of the Euclidean distance between their feature distributions $p_1(x)$ and $p_2(x)$. The Euclidean distance between the two pdfs is the integral of the squared difference over the whole feature space:

$$e = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} [p_1(x) - p_2(x)]^2 dx_1 \dots dx_N. \quad (5)$$

In [9] a closed-form solution for e was derived. The squared Euclidean distance is actually a measure of dissimilarity, thus the smaller the distance, the more similar the samples are.

The distribution of features is modeled here using GMMs. The parameters of the GMMs are estimated using the Expectation Maximization (EM) algorithm. It estimates the means and variances for a predefined number of components. The features used in the similarity estimation are listed in Table 12 at the end of the paper. The features are such that they attempt to describe properties of different types of audio signals, in order to work in a wide range of audio databases.

It is worth noticing that similarity estimation results also in implicit classification of audio database and hence efficiently extends the basic intuitive audio classification discussed in Section 2 towards unforeseeable audio databases.

6. QUERY-BY-EXAMPLE ALGORITHM

One of the most common operations which user makes in his personal database is searching for samples which have certain content. The purpose of query-by-example is to make the search easier for the user. An example sample can be given to the system describing the type of content the user is searching for. The system retrieves those samples from the database which, according to the algorithm used, are closest match to the example. The query-by-example application can also take advantage on both similarity estimation and audio classification. When the user provides an example to the system, the system can be directed to run the query first only on samples which are classified to the same class as the example. Among other things, this approach clearly has the benefit that the most promising query samples are fast delivered to the user.

However, if the database is large the search becomes exhaustive if the distance between example and all the samples in the database have to be calculated. Thus, we have applied key sample transformation and clustering algorithm [8]. The transformation from series of feature vectors to a k -dimensional feature space is required in order to effectively cluster the database but at the same time minimum amount of information should be lost.

The transformation used here is based on distances to the key-samples chosen from the database. The transformation is defined as follows:

$$T(x, O, d) = \Gamma \rightarrow \mathcal{R}^k, \quad (6)$$

where x is the original series of feature vectors, O is the set of k key-samples, d is the distance measure, Γ is the original feature space, and \mathcal{R}^k is the k -dimensional feature space in which i^{th} element is the distance from x to i^{th} key-sample ($i=1, \dots, k$).

The k samples are first chosen randomly from the database to work as key-samples. Then distances from each sample in the database to these key-samples are calculated and after the transformation the new feature vectors summarize distances from the sample to all of these key-samples. Finally, the database is clustered with the k -means algorithm using these new feature vectors.

The transformation and clustering can be made offline. In query, the nearest cluster to query sample is found using these random sample distances. Then the actual query with original series of feature vectors is first made inside the closest clusters by calculating the Euclidean distance between pdfs. The query can then be improved by widening the search to the further clusters.

The advantage of using this transformation is that we achieve significant speedup in clustering system, since instead of series of feature vectors, we can operate with single feature

vectors. Simultaneously, we are able to use more accurate distance measures in search, since in contrast to full search, only a small fraction of all combinations have to be calculated. The outline of the QBE algorithm is presented in Figure 5.

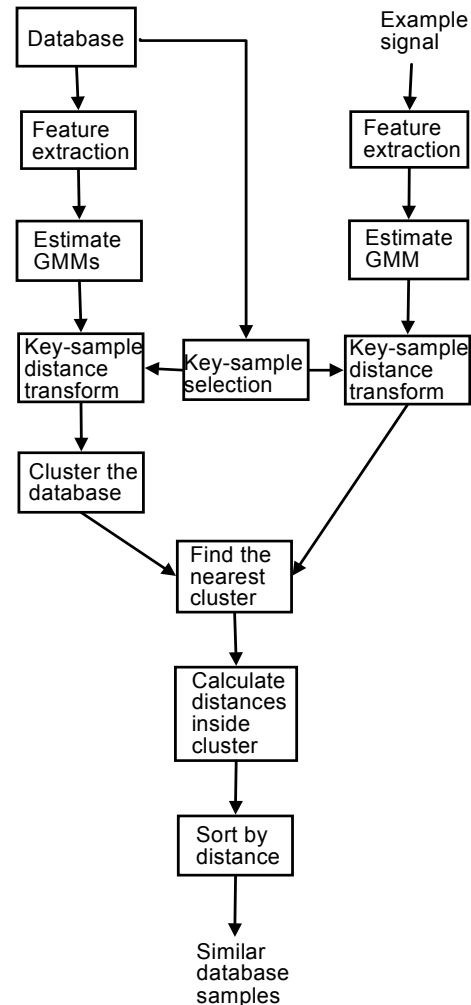


Figure 5: A block diagram for the query by example system.

Finally, after estimating the similarity between the example and the database samples, the decision have to be made, which samples are retrieved to the user as similar ones. There are two main principles for this, both having their own pros and cons. First is ϵ -range query in which the threshold for similarity is set, below which all the samples are retrieved. Second is k -nearest neighbor (k -NN) query in which the fixed number of most similar samples is retrieved.

In the optimal situation the ϵ -range query will retrieve all the samples that are considered similar, whereas the k -NN query always retrieves only k most similar samples. On the other hand, the ϵ -range query might end up retrieving zero samples if the variance between the samples in example class is higher than the average in the database but would still be considered similar. Respectively, if the variance between two classes is small enough, the ϵ -range query might retrieve samples from both classes. Furthermore, in the ϵ -range query the problem of finding a proper ϵ might be a complicated task and might require going through the whole database. However, ones the ϵ is set the similar samples can be retrieved already during the query. The drawback in the k -NN query is that the whole database must always be searched before accurate results can be retrieved. However, our clustering approach offers a significant speedup

compared to full search and in our simulations the k-NN query was used.

7. EXPERIMENTAL RESULTS

The experiments were performed individually for each component algorithm. Based on the results conclusions on the overall performance of the system can be drawn. The detailed combined results, however, are omitted here as presenting them for all the meaningful combinations would take too much space in the paper. The key in combining the algorithms to work together is to have as high performance for each component algorithm as possible.

For evaluating speaker change detection and similarity algorithm performances, we are using the precision *PRC*, recall *RCL* and the *F*-score measures. The precision and recall measures are defined as:

$$PRC = \frac{\text{number of correctly retrieved}}{\text{total number of retrieved}}, \quad (7)$$

and

$$RCL = \frac{\text{number of correctly retrieved}}{\text{total number of correct samples}}. \quad (8)$$

The evaluation of the segmentation quality is made in terms of *F*-score, a combined measure of *PRC* and *RCL* of change detection. *F*-score is defined as

$$F - score = \frac{2.0 * PRC * RCL}{PRC + RCL}. \quad (9)$$

The *F*-score values vary from 0 to 1, with a higher *F*-measure indicating better performance.

7.1 The test audio database

The database used in the tests simulates the user's personal database both in terms of quality and content. The database has been divided into three subsets. Selected subsets are used for directed testing of the SCD, GEND, EMOD, and SIMqbe algorithms. Audio analysis is performed on the whole database. In all the cases the 16 kHz sampling rate was used.

SCD subset:

The Dataset contains 99 separate in-house recordings, which contain equal amount of 2, 3, and 4 speaker recordings. The amount of different speakers in the dataset is 21 speakers representing both genders. Average duration of the speaker segment is 11.6 seconds and the number of speaker change points in total is 630.

EMOD subset:

The emotion database used in this study is the MediaTeam emotional speech corpus [18]. The database contains Finnish speech by 14 professional actors (8 male and 6 female) of ages 25-50 in the basic emotions (neutral, sad, angry, and happy). The data was divided into 280 sentence length samples.

Both SCD and EMOD subsets were used for GEND experiments.

SIMqbe subset:

The subset of the database used for testing the query-by-example algorithm contains samples from a wide range of

different audio events and environments. There are altogether 1529 audio samples. The samples were manually annotated into 3 main categories for AA tests and 17 sub categories for further similarity tests. In query-by-example experiments, the samples falling into the same subclass were considered to be similar. The classes and the number of samples in each class are listed in Table 1.

Main class	Sub class
Environmental (231)	Inside car (151) In restaurant (42) Traffic (38)
Music (785)	Acoustic (264) Drums (56) Electro acoustic (249) Symphony (51) Humming (52) Singing (60) Whistling (53)
Speech (316)	Speaker1 (50) Speaker2 (47) Speaker3 (44) Speaker4 (40) Speaker5 (47) Speaker6 (38) Speaker7 (50)

Table 1: Audio classes in the SIMqbe test database and the number of samples in each class.

7.2 Training databases and procedures

The training of the Bayesian network for audio classification was carried out with BNT Matlab toolbox. The training database contained following set of samples: Speech 264 min, Music 191 min, Silence 34 min, Constant Noise 99 min and Variable Noise 161 min. The training material was collected from completely different sources than the testing material used in this paper, thus the reported results represent a good estimation of the real life accuracy of the classifier.

7.3 Feature extraction and sharing

Having multiply algorithms supported in the system, there are usually subcomponents and even whole feature sets that are common at least to some of the algorithms. For example, the FFT-computation is costly but a common operation in almost all audio feature extraction. If there are several metadata extractors supported in the system, as in our case, considerable computational saving can be obtained by adopting the same analysis window length with all the algorithms and running the analysis only once. Of course, by forcing a single frame length for all the algorithms it has a potential degrading effect on the performance and it needs to be tested. In our case, however, the 30 msec analysis window was used and with the selected set of algorithms this effect was negligible. The only problem was caused by the EMOD and GEND algorithms for which the analysis window was too short for picking the pitch information reliably and a longer 60msec analysis window needed to be used. The feature set needed by the component algorithms in our work are listed in Table 12 at the end of the paper. The feature sets were selected on the basis of robustness and simplicity.

7.4. Audio classification results

The test results for the AA algorithm by using the whole test database are shown in Table 2. The table does not contain results for the silence class for natural reasons. The classification performance was, however, properly tested in our in-house tests and the AA algorithm did not have any difficulties even with the short pauses between words or during speaker change segments.

	Silence	Speech	Music	Env. Sounds
Speech	0.12	99.43	0.43	0.02
Music	0	20.25	78.98	0.76
Env. Sounds	0	0	19.05	80.95

Table 2: Confusion table showing the audio classification performance in percentages.

It can be seen that the classification rate for speech is already at a very high level. The classification rates, however, for music and environmental noise classes are compromised. Not surprisingly, most of the misclassifications occurred with the drum, humming, whistling, and sing samples, since the training database did not contain such sound samples

In a practical situation, it is not possible to be fully prepared against all the types of unforeseeable data but misclassifications are forced to happen. Later in the Section it is shown that this kind on errors can be well compensated and grouped by utilizing query-by-example on top of audio classification results. For now, we have settled to present the comparable results without the effect of the problematic samples in Table 3.

	Silence	Speech	Music	Env. sounds
Speech	0.12	99.43	0.43	0.02
Music	0	2.48	96.45	1.06
Env. sounds	0	0	19.05	80.95

Table 3: Confusion table showing the audio classification performance without drums, whistling, humming, and sing classes.

7.5. Speaker segmentation results

Speaker segmentation results are presented in Table 4. The results are compared to baseline, which is our earlier published implementation of the speaker segmentation that uses BIC profiles based false alarm compensation [3].

Method	<i>F</i> -score	Recall	Precision
Baseline	0.86	0.92	0.82
Proposed	0.90	0.92	0.89

Table 4: Speaker segmentation performance.

The results show that proposed speaker clustering algorithm merges efficiently the segments from the same speaker, while keeping the number of correctly detected changes in a high level. The relative improvement of *F*-score in segmentation results against the baseline result is 29.1%.

In Table 5, are presented speaker segmentation results for each number of speakers in a used test set. The number of speakers is unknown. It can be noted that *F*-score stays nearly at the same level in all cases.

Number of Speakers	<i>F</i> -score	Recall	Precision
2	0.92	0.91	0.92
3	0.89	0.90	0.90
4	0.88	0.95	0.84
All	0.90	0.92	0.89

Number of Speakers	<i>F</i> -score	Recall	Precision
2	0.92	0.91	0.92
3	0.89	0.90	0.90
4	0.88	0.95	0.84
All	0.90	0.92	0.89

Table 5: Speaker segmentation performance, the number of speakers is unknown.

In Table 6, are presented speaker segmentation results when the number of speakers is supervised. Often in practice it is difficult to get the information on the number of speaker in before hand. In some cases user might be willing to offer the information. Fortunately, based on the results, it can be seen that in both cases the performance is about the same.

Number of Speakers	<i>F</i> -score	Recall	Precision
2	0.92	0.93	0.91
3	0.89	0.90	0.89
4	0.89	0.91	0.87
All	0.90	0.92	0.89

Table 6: Speaker segmentation performance, the number of speakers is supervised.

Speaker clustering results are evaluated by comparing the manually annotated speaker labels with speaker labels from the speaker clustering algorithm. We calculated segmentation results using a script, which allows the reference and hypothesis speaker segments to have different labels, as mentioned in [19]. This may occur e.g. in situation when labelling detects falsely an extra speaker between speakers one and two. Speaker two becomes then speaker three, and all other segments from this speaker should be labelled as three even if the ground truth uses label two.

In Table 7 are presented the results for speaker clustering. Test was executed in two different ways. In a supervised manner the number of speakers was forced to be correct. For unsupervised tests the maximum number of speakers was set to a greater number than the real maximum number of speakers. It can be seen that the performances in both cases are close to each other.

Number of Speakers	Unsupervised	Supervised
2	94.73	97.61
3	86.22	89.29
4	84.61	87.90
All	88.52	91.60

Table 7: Speaker clustering results in terms of correct speaker label percentages.

7.6 Results for Emotion and Gender Classification

The classification results for the emotion detection algorithm using the EMOD database subset are summarized for reference below [20]. Tests were performed for three different scenarios in increasing difficulty (scenarios 1, 2, and 3). Also gender detection results using the same algorithm and the GEND database subset are provided (scenario 4). The results were obtained by using a standard forwards-backwards floating search feature selection algorithm [21] in conjunction with a kNN classifier using leave-one-out cross-validation to maximize the utilization of the data.

Scenario 1 – Speaker dependent case

This scenario assumes that the speaker, whose emotion is being evaluated, has been identified. Only the samples corresponding to the speaker are used as prototypes and an individual feature vector was searched for each subject. In this scenario, average classification performance of 96.1% was obtained for $k = 1$ with four features in each individual feature vector. The different feature vectors most commonly included features describing the distribution of F0 such as mean, max, and min F0-values.

Scenario 2 – Closed group case

In this scenario samples from other speakers are also included in the classification. In this case it is assumed that the speaker is a part of a predefined group, but has not been identified. Average classification performance of 85.7% was obtained for $k = 1$ with 13 features (see Table 12). The confusion matrix for scenario 2 is shown in Table 8.

	Neutral	Sad	Angry	Happy
Neutral	91.4	2.9	1.4	4.3
Sad	5.7	94.3	0.0	0.0
Angry	7.1	1.4	77.2	14.3
Happy	5.7	1.4	12.9	80.0

Table 8: Confusion matrix for emotion classification scenario 2.

Scenario 3 – True speaker independent case

In this scenario the samples of the speaker whose emotion is being evaluated are omitted from the training database. Only samples of the other speakers are used as prototypes. Average classification performance of 71.4% was obtained for $k = 5$ with 8 features (see Table 12). The confusion matrix for scenario 3 is shown in Table 9.

	Neutral	Sad	Angry	Happy
Neutral	82.9	5.7	4.3	7.1
Sad	14.3	72.8	4.3	8.6
Angry	10.0	1.4	64.3	24.3
Happy	14.3	4.3	15.7	65.7

Table 9: Confusion matrix for emotion classification scenario 3.

Scenario 4 – Gender classification case

A common feature vector for the GEND and EMOD data selection was searched for to attain an emotion insensitive gender classification setting. By including around 20% of emotional speech in the training the resulting feature vector is better able to handle both normal and emotional speech samples. An average performance of 96% correct gender classification was achieved using only a 3 dimensional feature vector (see Table 12) and a k value of 5. The result was further verified using the jointly calculated feature vector independently with both the SCD and EMOD selections. The SCD selection reached an independent average performance of 99% while for the EMOD selection the average performance was 92%. The slightly lower performance with dominantly emotional samples was rather unsurprising as the prosodic features of an emotional speaker clearly have higher deviations than in the neutral case. The confusion matrix for scenario 4 is shown in Table 10.

	Male	Female
Male	96.8	3.2
Female	4.7	95.3

Table 10: Confusion matrix for gender classification.

7.7. Query-by-example results

In query-by-example tests, the database is first clustered to 17 clusters using key-sample transformation and k-means clustering. Then one audio sample file is drawn from the database at the time to serve as an example and it is compared against the other samples which were clustered to the same cluster with the example. The 10 most similar samples are retrieved. The procedure is repeated for all the samples in the database. Table 11 represents the confusion matrix of the query results. The most confusion is between acoustic, electro acoustic, and symphony classes, which is understandable considering how similar these classes are also from the human perspective. The overall precision here is 91.1 %. The accuracy of full search is 94.1 %, which means that the effect of clustering is only 3 percent units in precision. However the speedup is directly proportional to the number of clusters.

It is interesting to observe how well based on the results the subclasses are separated from each other inside the same main class. Since it is practically impossible to predict the types of data people are collecting with their personal devices SIMqbe can be used for implicitly clustering the unforeseeable data within AA classes. This has much value for many applications.

Based on the results the SIMqbe algorithms captures well also the differences between the speakers. This makes SIMqbe algorithm valuable in creating relationship metadata together with the SCD algorithm as mentioned earlier.

8. CONCLUSIONS

State-of-art analysis tools for personal audio content management were discussed in this paper. On top of the general audio classification results three analysis algorithms were applied. Improved speaker segmentation and clustering algorithm was proposed. That was shown to provide comparable speaker clustering performance both in unsupervised (88.52%) and supervised (91.60%) use. For speaker segmentation F -score value of 0.90 was obtained in both cases. Gender detection was combined with the high performance emotion detection algorithm in order to take full advantage of prosodic feature computations and the detection algorithm. For emotionally neutral speech the gender detection rate was already extremely high. With our approach 96% gender detection rate was obtained also in case of emotional speech.

For compensating the problems with unforeseeable data and hence the absence of general analysis tools for the non-speech audio classes the use of efficient audio similarity measure and query-by-example algorithm with database clustering capabilities was proposed. Based on the test results it can be stated that the combined use for example with speaker segmentation and clustering algorithm to provide relationship metadata across all the database samples is justified.

Based on the work it can be stated that the above framework with some common computational configurations can be supported also in personal mobile devices while the combined results being sufficiently high for many personal audio content management purposes.

	Inside car	In restaurant	Traffic	Acoustic	Drums	Electro acoustic	Symphony	Humming	Singing	Whistling	Speaker1	Speaker2	Speaker3	Speaker4	Speaker5	Speaker6	Speaker7
Inside car	97.55	0	0.01	1.66	0	0.73	0	0	0	0	0	0	0	0	0	0	0
In restaurant	0.71	99.29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Traffic	5.79	3.68	86.58	0	0	0	0	0	1.05	3.68	0	0	0	0	0	0	0
Acoustic	0.04	0	0	88.56	0	8.98	0.68	0.19	1.1	0.11	0	0	0	0	0	0.34	0
Drums	0	0	0	0	96.79	1.79	1.43	0	0	0	0	0	0	0	0	0	0
Electro acoustic	0.08	0	0	11.85	0	86.75	0.64	0	0.32	0	0	0	0.28	0	0	0.08	0
Symphony	0	0	0	13.92	0.2	17.45	63.53	1.96	0.2	1.96	0	0	0	0	0	0	0.78
Humming	0	0	0	1.92	0	0	0	88.27	4.42	0	0	0	0	0	5.38	0	0
Singing	0	0	0.33	1.83	0	0.17	0.33	8	83.17	4.17	0.17	0	0	0	1.83	0	0
Whistling	0	0	0.38	0.38	0.19	0	1.51	0	3.96	93.58	0	0	0	0	0	0	0
Speaker1	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Speaker2	0	0	0	0	0	0	0	0	0	0	0	99.36	0.64	0	0	0	0
Speaker3	0	0	0	0	0	0	0	0	0	0	0	7.05	92.27	0	0	0.68	0
Speaker4	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Speaker5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
Speaker6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Speaker7	0	0	0	0	0	0	0	0	0	0	0	1.8	1.8	0	0	0	96.40

Table 11: Confusion matrix for query-by-example when 10 nearest neighbors were retrieved

9. REFERENCES

- [1] L. Lu, H.-J. Zhang and H. Jiang, "Content Analysis for Audio Classification and Segmentation", IEEE Trans. on Speech and Audio Processing, vol. 10, no 7, pp 504-516, 2002.
- [2] S. S. Chen, P.S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering Via the Bayesian Information Criterion", 1998 DARPA Broadcast News Transcription & Understanding Workshop, 1998.
- [3] O. Vuorinen, J. Peltola, S.-M. Mäkelä, "Unsupervised Speaker Change Detection for Mobile Device Recorded Speech", in Proc. IEEE ICASSP'07, pp. 757-760, Honolulu, USA 2007.
- [4] M. Naito, L. Deng and Y. Sagisaka, "Speaker Clustering for Speech Recognition Using Vocal-Tract Parameters", Speech Communication, vol. 36, no. 3, pp. 305-315, 2002.
- [5] R. Huang, J.H.L. Hansen, "Unsupervised Audio Segmentation and Classification for Robust Spoken Document Retrieval", IEEE ICASSP-2004, volume 1, pp. 741-744, May 2004.
- [6] J. Toivanen, E. Väyrynen, T. Seppänen, "Automatic discrimination of emotion from spoken Finnish", Language and Speech, 2004, 47 [4], pp. 383-412.
- [7] M. Helén, and T. Lahti, "Query by Example Methods for Audio Signals," In Proc. 7th Nordic Signal Processing Symposium (NORSIG'06), Reykjavic, Iceland 2006.
- [8] M. Helén, T. Lahti, "Query by Example in Large Databases Using Key-sample Distance Transformation and Clustering", in Proc. IEEE-MIPR'07, Taichung, Taiwan 2007.
- [9] M. Helén and T. Virtanen: "Query by Example of Audio Signals Using Euclidean Distance Between Gaussian Mixture Models," IEEE ICASSP'07, pp. 225-228, Honolulu, Hawaii, USA 2007.
- [10] S.-M. Mäkelä, J. Peltola, M. Myllyniemi. "Mobile Video Capture Targeted Narrowband Audio Content Classification", in Proc. ICASSP'06, France 2006.
- [11] E. Pampalk, "A Matlab Toolbox to Compute Music Similarity From Audio", in Proc. ISMIR'04, pp. 254-257, Spain 2004.
- [12] H. Murthy, S. Haykin, "Bayesian Classification of Surface-Based Ice-radar Images", Oceanic Engineering, IEEE Journal of, volume 12, issue 3, pp. 493 - 502, 1987.
- [13] P. Korpipää, M Koskinen, J. Peltola, S.M. Mäkelä, T. Seppänen, "Bayesian Approach to Sensor-Based Context Awareness", Personal and Ubiquitous Computing, vol. 7, no. 2, pp. 113 124, July 2003.
- [14] J. Ramírez, J. C. Segura, C. Benítez, A. de la Torre, A. Rubio, "Efficient voice activity detection algorithms using long-term speech information", Speech communication, vol. 42, pp. 271-287, 2004.
- [15] O. Vuorinen, T. Lahti, S.-M. Mäkelä, J. Peltola, "Light Weight Mobile Device Targeted Speaker Clustering Algorithm", Submitted to MMSP 2008.
- [16] K. R. Scherer, "Universality of Emotional Expression", In D. Levinson, J. Ponzetti & P. Jorgenson (Eds.), Encyclopedia of Human Emotions, pp. 669-674. New York 1999.
- [17] T. Seppänen, E. Väyrynen, J. Toivanen, "Prosody-based Classification of Emotions in Spoken Finnish", Proceedings of the 8th European Conference on Speech Communication and Technology, pp. 717-720, Geneva, Switzerland 2003.
- [18] T. Seppänen, J. Toivanen, E. Väyrynen, "MediaTeam Speech Corpus: A First Large Finnish Emotional Speech Database", in Proc. of the 15th International Congress of Phonetic Sciences, vol. 3, pp. 2469-2472, 2003.

[19] X. Anguera, J. Hernando, "Evolutive Speaker Segmentation using a Repository System", in Proc. of ICSLP'04, Jeju Island, Korea 2004.

[21] P. Pudil, J. Novovičová, J. Kittler, "Floating Search Methods in Feature Selection", Pattern Recognition Letters 15 (11), pp. 1119-1125, 1994.

[20] E. Väyrynen, "Automatic Emotion Recognition from Speech", University of Oulu, Department of Electrical and Information Engineering, Master's Thesis, 2005.

FEATURES	FEATURE DESCRIPTION	USED FOR
MFCCs	Mel Frequency Cepstral Coeffients	SCD, SIMI
Power variance	The variance of average log-power values from the past one-second interval	AA, SIMI
Low energy ratio (LER)	The ratio of frames with average power less than a predefined threshold (here 20%) of the mean of the frames in the past one second interval	AA, SIMI
Fluctuation Pattern Gravity (FPG)	Center of Gravity of the Fluctuation Pattern which describes the loudness of fluctuations in different frequency bands.	AA, SIMI
Harmonic ratio	The ratio of harmonic to the non-harmonic components (MPEG-7 stand.)	AA, SIMI
Lag	Fundamental frequency estimate from MPEG-7 harmonic ratio algorithm	AA, SIMI
Spectral spread	The deviation of the log-frequency power spectrum from centroid (MPEG-7 stand.)	AA, SIMI
Crest factor	The peak amplitude divided by the root mean square value of the frame	SIMI
Noise likeness	The correlation coefficient between the original spectrum and the spectrum convolved with a Gaussian impulse	SIMI
Frame energy	The total energy of the frame	SIMI
LFE500	Proportion of Low Frequency Energy under 500Hz	EMO, scen 2
Mean	Mean F0 frequency (Hz)	EMO, scen 2
MedianInt	Median RMS intensity (abs., dB)	EMO, scen 2
fracMin	5% value of F0 frequency (Hz)	EMO, scen 2, 4
Sratio	Ratio of silence to speech	EMO, scen 2, 3
fracMax	95% value of F0 frequency (Hz)	EMO, scen 2
GDfallav	Average F0 fall steepness (Hz/cycle)	EMO, scen 2
LFE1000	Proportion of Low Frequency Energy under 1000Hz	EMO, scen 2
GDnegav	Average F0 fall during continuous voiced segment (Hz)	EMO, scen 2
Median	Median F0 frequency (Hz)	EMO, scen 2, 4
GDrisemax	Max rise during continuous voiced segment (Hz)	EMO, scen 2, 3
MinInt	Min RMS intensity (abs., dB)	EMO, scen 2
Vratio	Ratio of voiced to unvoiced segments	EMO, scen 2
IntRange	Intensity range (abs., dB)	EMO, scen 3
fracRange	5% - 95% F0 frequency range (Hz)	EMO, scen 3
Shimmer	Trend corrected mean proportional intensity perturbation.	EMO, scen 3
Jitter	Trend corrected mean proportional F0 perturbation	EMO, scen 3
IntVar	Intensity variance (abs., dB)	EMO, scen 3
Norm_intvar	Normalized segment intensity distribution width variation	EMO, scen 3
GDriseav	Average F0 rise steepness (Hz/cycle)	GEND, scen 4

Table 12: List of various features used by the component metadata extractor algorithms.