On Enabling Techniques for Personal Audio Content Management

Tommi Lahti[1], Marko Helén[2], Olli Vuorinen[3], Eero Väyrynen[4],
Juha Partala[4], Johannes Peltola[3], Satu-Marja Mäkelä[3]

Nokia Research Center, Personal Content & Media Team, Tampere, Finland[1],
Tampere University of Technology, Department of Signal Processing, Tampere, Finland[2],
VTT Technical Research Center of Finland, Oulu, Finland[3],
University of Oulu, Oulu, Finland[4]

Contacts: tommi.lahti@nokia.com, marko.helen@tut.fi, olli.vuorinen@vtt.fi

ABSTRACT

State-of-the-art automatic analysis tools for personal audio content management are discussed in this paper. Our main target is to create a system, which has several co-operating management tools for audio database and which improve the results of each other. Bayesian networks based audio classification algorithm provides classification into four main audio classes (silence, speech, music, and noise) and serves as a first step for other subsequent analysis tools. For speech analysis we propose an improved Bayesian information criterion based speaker segmentation and clustering algorithm applying also 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. The experimental results show that the combined use of the algorithms is feasible in practice.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing - *Abstracting methods, indexing methods*.

General Terms

Algorithms, Experimentation, Performance.

Keywords

Personal audio content management, audio classification, speaker segmentation, emotion detection, query-by-example.

MIR'08, October 30–31, 2008, Vancouver, British Columbia, Canada.

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 other 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] and [8].

This paper 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. In Sections 4 and 5 the novel similarity measure and

the query-by-example algorithm is discussed. Section 6 summarizes the experimental evaluations. Finally, in Section 7 conclusions are made.

2. AUDIO CLASSIFICATION

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. Speech data in personal audio content management has natural and especially important role. The sub-categorization within the class is also much more intuitive than for the other classes. For example, speaker change detection, gender and emotion detection tools all rely on correct speech detection results.

We have slightly improved our earlier implementation [9] by incorporating fluctuation pattern features [10] 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.

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 [11]. Our network structure consists of N binary decision nodes $D_1, D_2, \ldots, D_N$ that are arranged hierarchically to separate the audio classes in a sequence into N+1 discrete classes $C_1, C_2, \ldots, C_{N+1}$. A set of audio features $F_i \in F = \{X_1, X_2, \ldots, X_K\}$ 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 $M_{\overline{C}_i}(F_i;\lambda_{\overline{C}_i})$, 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 $M_{\overline{C}_0}(F_0)$ =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

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

where $M_{\overline{C}_0}(F_0)$ =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 (see Table 6 at the end of the paper). The class probabilities in our network are conditionally independent to each other, thus the sum of all is not one and the results need to be normalized.

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 but inside the segment the 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 1.

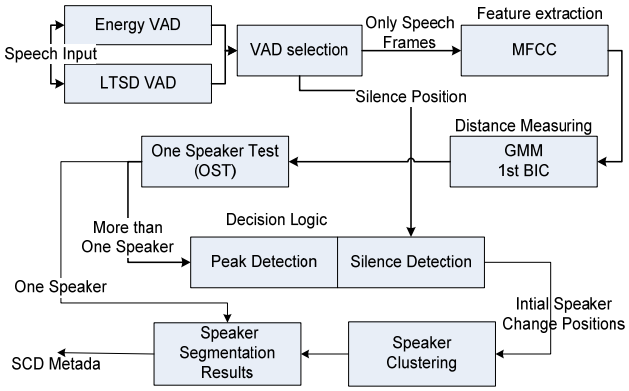

Figure 1: Block diagram of speaker segmentation system.

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 [12]. 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 in Section 3.2.

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, the output of the system is segmentation metadata. 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

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

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 x_1 and x_2. The variations between one speaker (one Gaussian) and two speakers (two different Gaussians) is given by

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

where λ is the penalty factor *and P* is the penalty term $P = ½ (p + ½p(p+1))$ x log N_X. p is the dimension of the acoustic space. 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 [3].

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 1. BIC matrix can be presented as:

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

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

In Figure 2 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 2 are: 1, 2, 3, 2, 3.

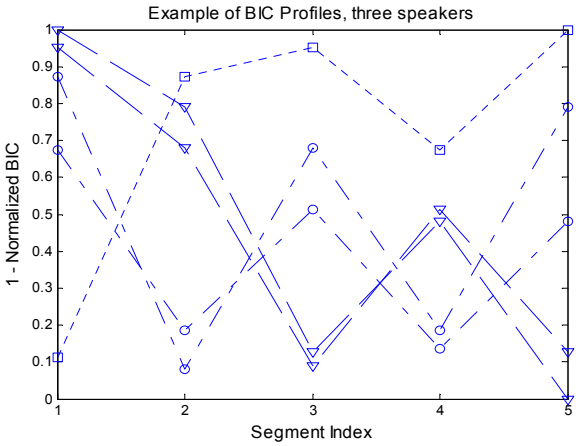

Figure 2: Example of BIC profiles. (square=speaker1, circle=speaker2, and triangle=speaker3).

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].

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 [14].

4. AUDIO SIMILARITY ESTIMATION

For similarity estimation, basically any distance measure can be used. In previous tests by Virtanen and Helén [8], 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 $p_2(x)$, which is actually a measure of dissimilarity, thus the smaller the distance, the more similar the samples are. The Euclidean distance between the two pdfs is the integral of the squared difference over the whole feature space:

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

In [9] a closed-form solution for e was derived. 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 6 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.

5. 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 [7]. 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:

$$\mathrm{T}(x, \mathrm{O}, d) = \Gamma \rightarrow \Re^k \,, \qquad (6)$$

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 $\Re^k$ is the k-dimensional feature space in which i^{th} element is the distance from x to i^{th} key-sample (i=1,...,k).

The k samples are 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 query-by-example algorithm is presented in Figure 3.

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. First is ε-range query in which the threshold

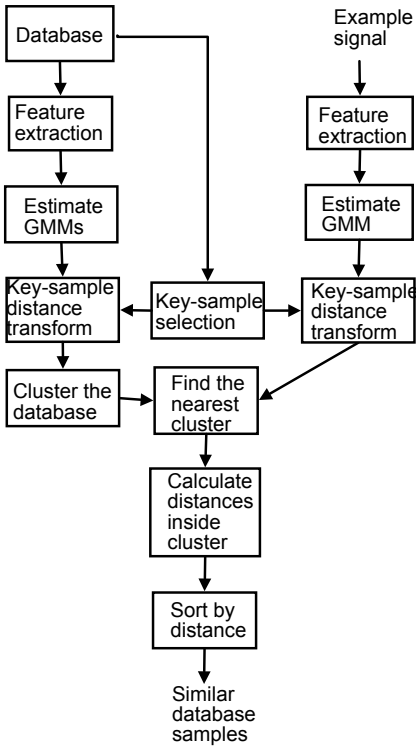

Figure 3: A block diagram for the query by example system.

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.

6. EXPERIMENTAL RESULTS

The experiments were performed individually for each component algorithm described above. In addition, our in-house system supports also emotion and gender detection. State-of-the-art gender detection can be obtained by utilizing the same algorithm as for emotion detection. Used setup is shortly summarized in this section together with a short description of the emotion classification procedure and the results. The algorithmic details and the performance figures for emotion detection have been published in [6] and [15] and are not presented here in detail due to space limitations. Based on the brief results, however, conclusions on the overall performance of the system can be drawn.

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

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

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

6.1 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, gender detection, and SIMqbe algorithms. Audio classification is performed on the whole database. In all the cases the 16 kHz sampling rate was used. In order to reduce the system complexity and to make the system run in a mobile device in a real-time a common analysis window length of 30 msec in preprocessing phase was used except for gender (and emotion) detection. Based on the in-house experiments this forcing approach does not degrade the performance of any of the algorithms considerably. The 30 msec analysis window was too short for picking the pitch information reliably and a longer 60 msec analysis window was used for gender detection.

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.

Emotional speech subset:

The emotion database used in this study is the MediaTeam emotional speech corpus. 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.

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.

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

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)

6.2 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. The features listed in Table 6 gave the best performance for the development database material. The development database was obtained by dividing the training material in two parts: training and testing.

6.3 Audio Classification Results

The test results for the audio classification 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 algorithm did not have any difficulties even with the short pauses between words or during speaker change segments.

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

	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

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.

6.4 Speaker Segmentation Results

Speaker segmentation results were compared against the baseline, which is our earlier published implementation of the speaker segmentation that uses BIC-profile based false alarm compensation [3]. The baseline *F*-score, *RCL* and *PRC* figures were 0.86, 0.92, and 0.82, respectively. The corresponding figures for the proposed algorithm were 0.90, 0.92, and 0.89, respectively. 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 3, are presented speaker segmentation results for each number of speakers in a used test set. The number of speakers is unknown. The *F*-score stays nearly at the same level in all cases.

Table 3: Speaker segmentation performance when the number of speakers is unknown.

#Speakers	*F*-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

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. Based on our tests, however, there is practically no performance gain in any case even if the number of speakers is known in advance.

Speaker clustering results are evaluated by comparing the manually annotated speaker label 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 [16]. This may occur e.g. in situation when labeling 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 labeled as three even if the ground truth uses label two.

In Table 4 are presented the results for speaker clustering. 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.

Table 4: Speaker clustering results in terms of correct speaker label percentages.

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

6.5 Gender and Emotion Detection Results

The algorithm utilized for emotion detection reaches an average emotional content discrimination performance of 71.4 % correct classification in a speaker independent case when using the basic emotions (neutral, sad, angry, and happy). Classification was performed using a standard sequential floating forwards-backwards feature selection algorithm [17] in conjunction with a kNN classifier. The classifiers were tested using a leave-one-out cross-validation method to maximize the utilization of statistical data. For detailed algorithmic description and a full description of the test results with our emotional data subset the interested reader should confer [6] and [15].

Table 5: Confusion matrix for query-by-example when 10 nearest neighbors were retrieved.

	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

The classifier setup was trained also for gender detection purpose. By utilizing the same algorithmic framework computation costs are reduced. The gender detection when using a feature vector trained only on neutral speech material but including some emotional samples to the database while testing gave 99.6 % and 86.1 % gender detection rates on neutral and on emotional speech, respectively. The gender detection using a feature vector (shown in Table 6) trained on both neutral and emotional speech material (80/20 proportion) gave 98.4 % and 90.7 % corresponding gender detection rates when using the same database configuration.

6.6 Query-by-Example Results

In query-by-example tests, the database was first clustered to 17 clusters using key-sample transformation and k-means clustering. Then one audio sample file was drawn from the database at the time to serve as an example and it was compared against the other samples which were clustered to the same cluster with the example and 10 most similar samples were retrieved. The procedure was repeated for all the samples in the database. Table 5 represents the confusion matrix of the query results. The most confusion was 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 was 91.1 %. The accuracy of full search was 94.1 %, which means that the effect of clustering was only 3 percent units in precision. However the speedup was 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 main audio classes. Note, that SIMqbe also improves the overall system experience by grouping the misclassified but somehow mutually similar samples inside the main classes

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.

7. CONCLUSIONS

State-of-the-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 extremely high. With our approach a 96% average gender detection rate was obtained also in the case of emotional speech. By default, in personal audio recordings the emotion of the speech is varying freely. Although the user or the application developer might not be interested in emotion detection the results show the importance of including emotional training data when using prosodic feature computations for gender detection if emotional speech can reasonably be expected.

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.

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.

8. 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, 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", IEEE ICASSP 2007, Honolulu, USA.

[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 4, 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, T. Lahti, "Query by example in large databases using key-sample distance transformation and clustering", IEEE-MIPR'07, Taichung, Taiwan 2007.

[8] M.Helén and T.Virtanen, "Query by example of audio signals using Euclidean distance between Gaussian mixture models," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007), April 15-20, 2007, Honolulu, Hawaii, USA.

[9] S.-M. Mäkelä, J. Peltola, M. Myllyniemi. "Mobile video capture targeted narrowband audio content classification", in Proc. ICASSP'06, France 2006.

[10] E. Pampalk, "A Matlab Toolbox to compute music similarity from audio", Proc. 5th International Conference on Music Information Retrieval (ISMIR'04), pp. 254—257, Spain 2004.

[11] H. Murthy, S. Haykin, "Bayesian classification of surface-based ice-radar images", Oceanic Engineering, IEEE Journal of, volume 12, issue 3, pp. 493 – 502, Jul. 1987

[12] 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

[13] T. Wu, L. Lu, H.-J. Zhang. "UBM-based real-time speaker segmentation for broadcasting news", in Proc. ICASSP'03, Hong Kong 2003.

[14] O. Vuorinen, T. Lahti, S.-M. Mäkelä, J. Peltola, "Light weight mobile device targeted speaker clustering algorithm", Submitted to MMSP 2008.

[15] E. Väyrynen, "Automatic emotion recognition from speech", University of Oulu, Department of Electrical and Information Engineering, Master's Thesis, 2005.

[16] X. Anguera, J. Hernando, "Evolutive speaker segmentation using a repository system", in Proc. of ICSLP'04, Jeju Island, Korea 2004.

[17] P. Pudil, J. Novovičová, J. Kittler, "Floating search methods in feature selection", Pattern Recognition Letters 15 (11), pp. 1119-1125, 1994.

Table 6: List of various features used by the component metadata extractor algorithms.

FEATURES	FEATURE DESCRIPTION	USED FOR
MFCCs	Mel Frequency Cepstral Coefficients	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
Mean	Mean F0 frequency (Hz)	GEND
fracMin	5% value of F0 frequency (Hz)	GEND
GDriseav	Average F0 rise steepness (Hz/cycle)	GEND