A GENERIC SYSTEM FOR AUDIO INDEXING: APPLICATION TO SPEECH/ MUSIC SEGMENTATION AND MUSIC GENRE RECOGNITION

Geoffroy Peeters
IRCAM - Sound Analysis/Synthesis Team, CNRS - STMS
Paris, France
peeters@ircam.fr

ABSTRACT

In this paper we present a generic system for audio indexing (classification/ segmentation) and apply it to two usual problems: speech/ music segmentation and music genre recognition. We first present some requirements for the design of a generic system. The training part of it is based on a succession of four steps: feature extraction, feature selection, feature space transform and statistical modeling. We then propose several approaches for the indexing part depending of the local/ global characteristics of the indexes to be found. In particular we propose the use of segment-statistical models. The system is then applied to two usual problems. The first one is the speech/ music segmentation of a radio stream. The application is developed in a real industrial framework using real world categories and data. The performances obtained for the pure speech/ music classes problem are good. However when considering also the non-pure categories (mixed, bed) the performances of the system drop. The second problem is the music genre recognition. Since the indexes to be found are global, "segment-statistical models" are used leading to results close to the state of the art.

1. INTRODUCTION

Automatic audio indexing has become a major concern today. Given the increasing amount of audio indexing applications (sound recognition, music genre/ mood recognition, singer type recognition, speaker recognition, speech/ music segmentation...) many different applications have been, are and will be developed. However most of these applications rely on the same underlying concepts: extract a set of time-frame feature vectors, train a statistical model using hand-labeled data in order to create a "classifier" and finally use this classifier to label unknown data. Because of that, developing a unique generic and modular indexing system is attractive.

In the ongoing French national project "Ecoute", two of these indexing applications are to be developed: a speech/ music segmentation and a music genre recognition system. It has therefore been decided to develop this generic audio indexing system and apply it to the two problems. The goal of this paper is to present this generic indexing system and detail its application to the two problems.

Several generic systems have been proposed so far. For example, the Waikato University WEKA [1] system is a generic machine learning system written in Java. However its direct applicability to the audio case is not obvious (no feature extraction, no consideration of time information). The Sony EDS [2] system performs both feature extraction and machine learning but is heavy in computation time. The McGill University jAudio[3] + ACE[4], Tzanetakis' Marsyas[5], or IMIRSEL's M2K [6] systems all seem promising solutions but it still need to be proven that they provide large performances for specific applications.

Our generic system is based on a system we previously developed for a task of instrumental sound recognition[7]. For this task the system showed very good performances. The training stage of the system is based on a succession of four steps: feature extraction, feature selection, feature space transform and statistical modeling. The system has been modified and extended to make it generic and modular. The requirements for the design of such a generic system are presented in part 2.1. The system we have developed is presented in part 2.2. We then present the results of applying it to the two considered problems: speech/ music segmentation (part 3.1) and music genre recognition (part 3.2).

2. GENERIC AUDIO INDEXING SYSTEM

2.1. Requirements for a generic system

The two main actions the system must perform are training and indexing. "Training" denotes the stage in which a classification model is learned from hand-labeled data. "Indexing" denotes the stage in which this classification model is used to label or segment unknown data. The two actions must be clearly separated since they are not used by the same people.

Training consists in extracting features from a set of audio files (or a database) and finding a mapping between the characteristics of the features and hand-annotated labels of the audio files. An audio file can have a unique label (for example a "music genre" label describes a whole music track file) or a succession of labels over time (a 24h radio program file is described by a succession of labels over time: speech, music, jingles...). These labels define the problem to be solved. The set of files and the corresponding labels must be easy to define and modify by the user.

The performances of the system depend strongly on the features used. Each problem may require different features. Therefore, in a generic system, changing the feature extraction stage must be easy. Conversely the system must be able to choose by itself the appropriate subset of audio features in order to solve a specific problem.

The performances of the system also depend strongly on the choice of the model used to represent the mapping between the features and the labels (the classification model). SVM is known to perform the best but is limited to two classes problems, KNN also perform very well but is limited by the size and dimensionality of the training set. This part of the system must also be easily parametrizable.

Part of the training consists in testing the performances of the trained model. Several evaluation methods can be used for that:

cross-database validation, N-folds cross validation or Leave-One-Out validation.

The system we have developed takes the previous requirements into account. The global flowchart of the system is presented in Fig. 1. We describe it in the following part.

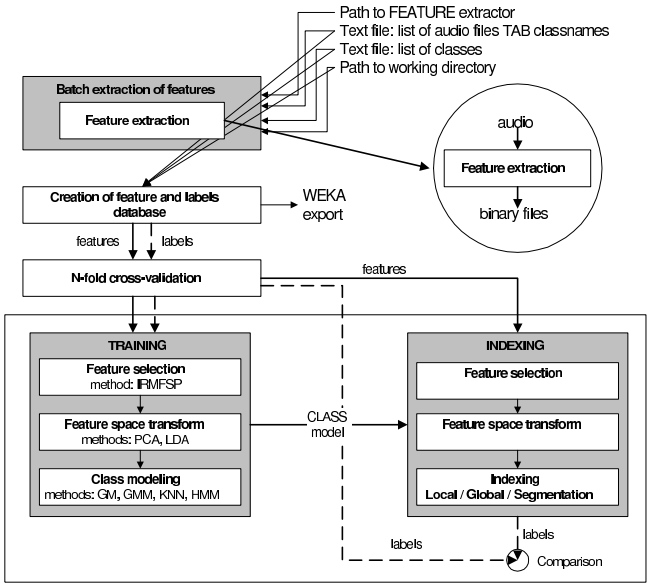

Figure 1: *Flowchart of our generic indexing system.*

2.2. Description of the system

2.2.1. Describing a new indexing problem

In order to make the description of a new indexing problem easy for the user, we have chosen to use a simple set of text files.

List of audio files and annotations: The first text file contains a list of audio files that will be used by the system to learn the characteristics of the classes. Each row of the text file contains the path to an audio file followed by the name of the corresponding annotated class. The user also has the possibility to replace the name of the annotated class by the path to a Wavesurfer[8] .lab file which allows to annotate, for the same audio file, a succession of classes over time.

List of classes and mappings: The second file contains the list of the name of the classes that will be considered for the training. This list can be a subset of the classes used for the annotation of the audio files (we call the later "annotated classes"). In this case, only the files (or the temporal segments) corresponding to this subset will be considered during the training. This file can also perform a mapping between the annotated classes and new class names. This allows mapping annotated class names between various databases. Several annotated classes can also be mapped to the same new class. This allows creating hierarchy among classes. For example combining the annotated "talk-voice" and "ads-voice" classes (see part 3.1) into a unique trained "speech" classes is very easy with our system.

Extractor: Finally, the last input to the system is the path to a "feature extractor". A "feature extractor" is a program that takes as input the path to an audio file and output the features values over time in a binary file. The format of the output file is self-defined in order to 1) make it usable without the knowledge of the feature extractor, 2) gives the necessary information in order to guarantee that all the feature files used by the system are compatible. In order to do that, each file contains the feature values, feature names, an identifier to the used feature extractor, the parameters of it and its version.

2.2.2. Training the system

In order to train the system, we first extract all the features of the audio files to be used as examples to learn the classes. This is done in a batch process using all the files defined in the list of audio files and using the defined "feature extraction" program. The results of this is a set of binary feature files.

A database is then created containing all the feature and class values. For this, the system reads all the binary files containing the features over time, reads the class definitions and perform the mapping between the features and the classes. At this stage, the user can export all the data in the Weka[1] format in order to perform external statistical analysis.

The training of the system then starts. It is a succession of three stages.

Feature selection: The first stage of the system selects among all extracted features the ones that are the most useful to describe the classes. The algorithm currently used is the Inertia Ratio Maximization with Feature Space Projection (IRMFSP) we proposed in [7].

The IRMFSP algorithm measures for each individual feature (we consider a multi-dimensional feature, such as the MFCC, as a set of scalar features) the inertia ratio (also named Fisher discriminant) knowing the feature values and their class belonging. The algorithm then selects the feature with the largest inertia ratio (the most discriminative feature). It then applies an orthogonalization process by projecting the whole feature space on the selected feature. This process guarantees that the remaining features are orthogonal to the selected feature (i.e. no more correlated). The process is then repeated in order to select the next features.

Feature space transform: The second stage transform the feature space in order to reduce its dimensionality while improving class representation. Currently two transforms are used: -the PCA which reduces the dimensionality of the feature space while preserving most of the variance of the data, -the LDA which reduces the dimensionality while maximizing the class separation of the data.

Class modeling: Finally, the third stage performs the statistical modeling of the problem. The following models are currently available: multi-dimensional Gaussian modeling, Gaussian mixture modeling, K-Nearest Neighbors, hidden Markov models, various unsupervised clustering algorithms, and histogram learning.

The output of the training is a "CLASS model" file, which stores all the parameters of the training and the references to the extractor to be used. This file can then be used for the indexing of unknown files.

2.2.3. Indexing

The current system can process the two following types of indexing:

Local indexing means that various labels are assigned over the file duration. In this case, each time frame (feature vector) is processed separately and classified separately. Smoothing techniques over time can be applied by computing short-term histogram of class belonging or by applying median filtering. Class changes over time can then be used to perform segmentation and assign a label to each segment. The local indexing will be used in part 3.1 for the speech/ music segmentation problem.

Global indexing means that a single global label is assigned to the whole file (or segment) duration. This is the case when

1. the feature vector is timeless, i.e. it describes directly the whole file (or segment) duration (this was the case in our instrumental sound classifier[7]),
2. when a global decision is taken from a succession of instantaneous features. This is the case when using hidden Markov modeling or when using the methods proposed below.

The results of the indexing process is output in a simple text file (Wavesurfer format).

2.2.4. Global indexing methods

In a standard classification system, each feature-vector at each time frame $f(t)$ is considered independently as an observation of a specific class c_i. The training and indexing are therefore performed directly on a frame basis. We call this model a "frame-statistical model".

When all the frames of a given file (segment) are supposed to belong to the same class, one can benefits from this knowledge by performing a "vote" among the frame-classes. This allows improving the accuracy of the classification by reducing the effect of local (frame-class) misclassification. We present below the "cumulated histogram" and the "cumulated probability" method.

In some cases (as for example the music genre problem of part 3.2), a segment (file) belongs as a whole to a class-to-be-found, but its individual frames do not necessarily belong to the class-to-be-found (in the case of music genre, a given time frame of a rock song can be very close acoustically to a given time frame of a blues song, therefore the various time frames of a given track could belong to various classes while the whole track belongs to only one class). The class-to-be-found is rather defined by a specific distribution or succession over time of frame-classes. The "segment-statistical model" presented below allows to take this into account.

"Cumulated histogram": The decision about the global class of a file/ segment is made by choosing the class with the largest number of occurrences among the frames. For this, each frame of the file/ segment is first classified separately: $i(t) = \arg\max_i p(c_i|f(t))$. The histogram $h(i)$ of class belonging $i(t)$ over all the frames is then computed (the bins of the histogram corresponds to the various classes c_i). The class corresponding to the maximum of the histogram $h(i)$ is finally chosen as the global class. In the following we call this method "cumulated histogram".

"Cumulated probability": Another possibility, is to use the frame probabilities $p(c_i|f(t))$, cumulate them over all the frames belonging to the file/ segment ($p(c_i) = \frac{1}{T}\sum_t p(c_i|f(t))$ and choose the class i with the highest cumulated probability ($i = \arg\max_i p(c_i)$). We call this method "cumulated probability".

"Segment-statistical model": In this paper, we propose to learn the characteristics of the "cumulated probability" and use the corresponding statistical models to perform the classification. We note s a specific segment (file) and $p_s(c_i)$ its cumulated probability. We note S_i the set of segments (files) of the training set belonging to a specific class i. For each class i, we compute the set of "cumulated probabilities" $p_{s\in S_i}(c_i)$. For a specific class i, we then model the behaviors of the bins c_i over all the $s \in S_i$. We call this model a "segment-statistical model" and note it $\hat{p}_i(c_i)$. In order to index an unknown segment/ file, we first compute its "cumulated probability" $p_s(c_i)$ and classify it using the trained "segment-statistical models" $\hat{p}_i(c_i)$.

Two statistical models have been considered:

1. The first one uses a Gaussian modeling of the bins of c_i. For each class i, we compute the mean and standard deviation of each bin c_i over all the $s \in S_i$.
2. The second model uses only the above-mentioned mean values. In this case, the indexing is performed by choosing the class i corresponding to the model $\hat{p}_i(c_i)$ with the largest cosine distance with the "cumulated probability" $p_s(c_i)$ of the unknown segment (file) s.

2.2.5. Validation

The current system can perform two types of validation.

Cross-database validation: one database is used for training the system, another one is used to evaluate the performances of it.

N-folds cross validation: the set of data is divided into N-folds: N-1 of them are used for training the system, the remaining one is used to evaluate the performances of it. In this case, special care must be taken in order to guarantee independence between the folds used for training and the one used for evaluation. In our system, we use the folder information of the files to detect dependencies. A specific case of N-folds cross validation is the Leave-One-Out validation in which N equal the number of data.

In part 3.1, we will used both validation methods, in part 3.2, only cross-database validation will be used.

2.2.6. Features

So far, two feature extractors have been developed.

Dedicated audio features: The first extractor we have developed is dedicated to the problem of instrument sound recognition and is described in details in [9]. In this extractor, the assumption is made that the audio file contains a single instrument note. Therefore the extraction of high-level concepts (such as attack time or fundamental frequency of a note) is *-feasible* (i.e. can be done considering current signal processing capabilities) and *-meaningful* (i.e. has a meaning for the given signal).

Generic audio features: In the case of generic audio (music, radio stream...) the extraction of such concepts would be *-difficult* (requiring either temporal segmentation or source separation) and *-meaningless* (considering that a 24h radio program or a music track has more than one attack time or release time). Therefore the second extractor we have developed only contains the subset of features that do not rely on any time model (such as the temporal organization assumption necessary to derive the attack time) or signal model (such as the harmonic sinusoidal model necessary to derive the fundamental frequency). It extracts instantaneous features such as MFCC and Spectral Flatness Measure.

2.2.7. Temporal modeling of features

Instantaneous features are usually extracted using a 40ms window with a hop size set to 20ms. This can lead to a very large amount of data for the training: 4 millions feature vectors for a 24 hours radio program file. In order to decrease the amount of data, "temporal modeling" of the feature vectors can be performed.

"Temporal modeling" means modeling the evolution over time of each feature using frame analysis. The length of the sliding window is typically chosen between 500ms to 2s and the modeling is performed over each window. The current system can perform the following type of "temporal modeling": statistical measures (mean, variance values over each window), histogram of cluster belonging, spectral decomposition of feature evolution [10] and sub-band grouping of this spectral evolution [11] [12].

3. TWO APPLICATIONS OF THE INDEXING SYSTEM

3.1. Speech/ music segmentation

The first application we present is a tool for automatic segmentation of radio streams. This tool is developed in coordination with a company that produces managing and archiving softwares for radio stations. The categories to be indexed as well as the radio corpuses are directly defined and provided from their clients and are thus real world categories and data.

Speech/ music segmentation has been the subject of a large amount of research in the last two decades. The front-end of most systems starts by extracting low-level features such as the zero-crossing rate, 4 Hz energy modulation, spectral moments, MFCC, entropy... Each class is then modeled using instance-based classifier (KNN), Bayesian classifier (Gaussian mixture model), Support Vector Machine... The field is well established and has dedicated evaluation protocols such as DARPA in the USA or ESTER [13] in France. We refer the reader to [14], [15], [16], [17], [18], [19] for major or recent publications in this field.

The goal of this part is twofold: first we want to test the applicability of our system for a task of speech/ music segmentation, secondly we want to test such a system in a real industrial framework.

3.1.1. Considered categories

Two sets of categories are to be found. The first set corresponds to *acoustical categories*: music, voice, mix and bed.

- "Mix" denotes segments where music and speech exist but do not overlap continuously over time; they rather succeed each other over time.
- "Bed" denotes segments where speech and music overlap regularly over time; a typical example of it is the introduction of news on the radio.

The second set of categories corresponds to *industry categories*, i.e. the categories used by radio programmers to annotate their programs: music, talk, ads and jingles. Obviously the categories "talk" and "ads" can be composed of voice, mix or bed. The category "jingle" can also be composed of any of the previous acoustical categories. We do not detail the "jingle" part of the system in this paper since it is processed by a dedicated audio finger-print system which allows to identify them. The correspondence between the industry and the acoustical categories is indicated into Tab. 1.

3.1.2. Corpus

The corpuses used for the development and testing of the system are the following:

Corpus RadioFrance: the speech part is composed of a subset of the MPEG-7 corpus made of recordings of Radio-France radio station in July 98, the music part is composed of two subsets: the ISMIR2004 "song excerpts" test set and a private music genre database.

Corpus UK: consists of 24h of recording of a major commercial radio group in the UK. This station has a high rate of audio compression, includes many ads, jingles, talks and music.

Corpus SUD: consists of 24h of recording of a regional radio station in France.

Each corpus has been annotated into the above mentioned categories. However, the RadioFrance corpus is only annotated in the categories music-music and speech-clean (equivalent to talk-voice and ads-voice). The annotations are in the Wavesurfer format. The distribution of the corpuses is indicated into Tab. 2. For each category we indicate its percentage (%) and its duration in minutes (m). As one can see, all three corpuses are highly unbalanced in favor of the music category.

3.1.3. System configuration

Features: For each corpus, we have extracted the following set of instantaneous audio features:

- 13 Mel-Frequency-Cepstral-Coefficients (using 40 triangularly-shaped Mel bands, and keeping the DC component),
- Delta-MFCC,
- Delta-Delta-MFCC,
- 4 Spectral-Flatness-Measure coefficients (the 4 rectangularly-shaped frequency bands are [250, 500], [500, 1000], [1000, 2000] and [2000, 4000] Hz),
- Delta-SFM,
- Delta-Delta-SFM.

The signal is first converted to mono and down-sampled to 11KHz. The frame analysis was performed using a 40ms Blackman window, the hop size was 20ms. We then apply temporal modeling to the 50Hz feature vector signal using the mean and variance values over a 2s window with a hop size of 1s.

Classifier: Various configurations of the classifier have been tested (variation of the number of selected features, choice of the statistical model, variation of the number of mixtures in the GMM...). The best results were obtained with the following configuration:

- Feature selection: IRMFSP algorithm using the first 40 selected features,
- Feature space transform: Linear Discriminant Analysis,
- Class modeling: GMM with 20 mixtures and full-covariance matrix. Since the corpus is highly unbalanced we did not use the prior probabilities in the Bayes formulation.

Industry / Acoustical	Music	Jingle	Voice	Mix	Bed
Music	music-music				
Jingle		jingle-jingle			
Talk			talk-voice	talk-mix	talk-bed
Ads			ads-voice	ads-mix	ads-bed

Table 1: *Correspondence between industry and acoustical categories for speech/music segmentation*

Corpus name		RadioFrance		UK	SUD
Description		french speaking		english speakin	french speaking
Total duration		622m		1375m	1333m
Classes:	music-music	74% (460m)	music-music	57% (788m)	70% (945m)
	speech-clean	26% (162m)	talk-voice	16% (222m)	23% (312m)
			talk-mix	8% (111m)	1% (13m)
			talk-bed	3% (41m)	1% (10m)
			ads-voice	4% (51%)	1% (10m)
			ads-mix	6 (89m)	3% (35m)
			ads-bed	5% (70m)	1% (8m)

Table 2: *Distribution of the three corpuses for speech/music segmentation*

		Found						
		'music-music'	'talk-voice'	'talk-mix'	'talk-bed'	'ads-voice'	'ads-mix'	'ads-bed'
Real	'music-music'	79,4	0,5	1,7	2,9	0,9	8,5	6,1
	'talk-voice'	0,5	71,8	8,1	5,0	12,4	1,3	0,8
	'talk-mix'	2,6	8,3	42,6	22,2	6,3	9,1	8,9
	'talk-bed'	4,1	3,9	34,9	39,8	5,3	6,4	5,6
	'ads-voice'	1,1	10,0	5,8	3,1	66,2	9,2	4,4
	'ads-mix'	12,1	2,3	9,4	5,8	10,4	41,7	18,3
	'ads-bed'	6,8	0,8	6,0	5,3	6,1	14,4	60,6

57,5

Table 3: *Ten-fold cross-validation confusion matrix of the speech/music system for the 7 categories problem using the UK corpus*

3.1.4. Results

7 classes problem: We first present the results obtained when considering blindly the 7 classes problem (blindly means that we do not take into account the fact that some classes are in fact acoustically equivalent): music-music, talk-voice, talk-mix, talk-bed, ads-voice, ads-mix and ads-bed. The results obtained using a ten-fold cross-validation method for the UK corpus (the most difficult) are indicated in Tab. 3. The average Recall (average over the classes) is $\overline{R} = 57.5\%$ (the random Recall for 7 classes would be $\overline{R} = 14.28\%$). Music-music and talk-voice are recognized at $R = 79.4\%$ and $R = 71.8\%$ respectively. The largest confusions occur with the non-pure categories (mix and bed) and when trying to distinguish talk from ads. The category talk-voice is mainly confused with ads-voice/ talk-mix/ talk-bed, the category talk-mix with talk-bed/ ads-mix/ ads-bed, the category ads-voice with talk-voice...

Mean Recall (Mean F-Measure) Music Recall - Speech Recall		Evaluation: Radio-France	Evaluation: UK	Evaluation: SUD
Training	Radio-France		**86,5 (87,9)** 99 - 73,9	**95,2 (96,4)** 90,9 - 99,5
	UK	**92,1 (57,6)** 84,3 - 99,9		**89,4 (92,7)** 79,1 - 99,8
	SUD	**95 (78,1)** 96,9 - 93,2	**90,2 (91,3)** 99,1 - 81,3	

Table 4: *Cross-database evaluation of the speech/music system for the 2 categories system using the three corpuses*

2 classes problem: We now only consider the pure categories and merge the acoustically equivalent categories. This leads to two classes: music-music and a category merging the categories talk-voice and ads-voice, which we call speech. For the UK corpus, using a ten-fold cross-validation method, the mean Recall is $\overline{R} = 95.6\%$ ($R_{music} = 96.7\%$ and $R_{speech} = 94.4\%$), for the Radio-France corpus it is $R_{music} = 96.48\%$ and $R_{speech} = 96\%$, for the SUD corpus it is slightly lower: $R_{music} = 95.8\%$ and $R_{speech} = 92.1\%$. Whatever the considered number of classes or the considered corpus, music tends to be more easily recognized than speech.

Cross-database validation: We now want to test the generability of the trained classification model. In particular, we want to test if the system has learned the general characteristics of music and speech or the specific characteristics of music and speech as played on the specific radio station used for training. In order to test this, we perform a cross-validation over the three corpuses: one corpus is used for training, the two remaining ones for evaluation. The results are indicated in Tab. 4. Each cells report the mean (over the classes) Recall $\overline{R}$, mean F-measure $\overline{F}$, and the music and speech Recalls. In the following, we note $R^{x \to y}$ the Recall obtained when training the model on the corpus x and using it to classify corpus y.

The best result is obtained when training the model using the RadioFrance corpus and applying it for the indexing of the SUD corpus: $\overline{R^{RF \to SUD}} = 95.2\%$; the second best result when using SUD to classify RadioFrance: $\overline{R^{SUD \to RF}} = 95\%$. The worst results are obtained when using RadioFrance to classify UK or UK to classify SUD. RadioFrance and SUD seem very close acoustically while UK seems very different. The assumption that the difference comes from the language of the corpuses (French/ English) is contradicted by the individual class Recalls. Actually, using UK to train the speech model and applying it to the RadioFrance or SUD corpuses leads to the highest Recalls: $R_{speech}^{UK \to RF} = 99.9\%$ and $R_{speech}^{UK \to SUD} = 99.8\%$ respectively. The difference between the corpuses seems to come mainly from the music part: $R_{music}^{UK \to RF} = 84.3\%$ and $R_{music}^{UK \to SUD} = 79.1\%$. The music of RadioFrance or SUD tends to be recognized as the speech learned from UK. Also the speech of UK tends to be recognized as the music of RadioFrance or SUD ($R_{speech}^{RF \to UK} = 73.9\%$ and $R_{speech}^{SUD \to UK} =$

81.3%). The music model is better trained using the SUD corpus: $R_{music}^{SUD \rightarrow RF} = 96.9\%$ and $R_{music}^{SUD \rightarrow UK} = 99.1\%$.

Comments on the Precision and F-measure: The values of the F-measure, or the Precision factor, must be analyzed with care since they strongly depends on the distribution of the test set which is highly unbalanced in our case. For example in the case of RadioFrance classified by UK, we get 99.9% Recall for the class "speech", but its Precision is only 13.4%. This looks like a large part of "music" has been classified as "speech". In fact this part is small in comparison to the number of "music" data: only 15.6% of the music data have been classified as speech. But since the total number of music data (m=48382 data) is much larger than the total number of speech data (s=1175), even 15.6% (0.156*m=7581) makes the Precision drops drastically (the Precision is computed as 0.999*s/(0.156*m+0.999*s).

Conclusion: Considering that no specific modifications of our system have been made for the specific task of speech/ music indexing, the results obtained for the two-classes problem are very encouraging. The choice of the training set seems however to be important for the generalization of the system and different corpus may be required for training the music and speech models. However, the application of our system for the seven-classes problem (including the non-pure categories "mix" and "bed") requires further development. In this case, the use of generic audio features (as used in our experiment) does not allow distinguishing efficiently the non-pure classes.

3.2. Music genre recognition

The second application we present is a tool for the automatic recognition of music genre. Although music genre categories have been showed to be fuzzy or hill-defined [20], their automatic estimation is a usual step in the understanding of the acoustical characteristics underlying music similarity. For this reason, it has been the subject of many contributions in recent years. Moreover dedicated frameworks, [21] or MIREX [22], are devoted to its evaluation which allows the comparison of newly developed algorithms to state-of-the-art algorithms. In opposition to speech/ music front-ends, two main categories of systems exist in the case of music genre recognition. The first one learned the classes directly from low-level features [23] [24] (MFCC, Spectral Contrast, Loudness, Roughness...). The second one learned the classes from high-level features [25] (tempo, beat histogram, chroma, pitch contours). Our system belongs to the first category since it uses the same set of low-level features as our speech/ music segmentation system. We refer the reader to [26] for a recent overview of the music genre topic.

3.2.1. Corpus and categories

For the evaluation of the performances of our system, we have used the test sets from the ISMIR2004 music genre contest [21]. It should be noted that we had only access to the training and development set, not to the evaluation one. Training of our system is done on the training set and the performances are given on the development set. The distribution of both sets are indicated in Tab. 5. As for the speech/ music corpuses, the corpus is here also very unbalanced in favor of the classical music category. The definition of the classes is also controversial: jazz and blues are merged into a single class, the "world music" class contains many different types of music.

3.2.2. System configuration

Features: The same set of features as for the speech/ music segmentation system has been used. However the modeling length was set to 4s (instead of 2s) and the hop size to 2s (instead of 1s).

Classifier: Various configurations of the classifier have been tested and the best results were obtained with the following configuration:

- Feature selection: no
- Feature space transform: Linear Discriminant Analysis,
- Class modeling: GMM with 5 mixtures and full-covariance matrix. Since the corpus is highly unbalanced we did not used the prior probabilities in the Bayes formulation.

3.2.3. Results

Since we know that all the frames of a given file belong to the same music track and should therefore have the same music genre class, we use the global indexing methods proposed in part 2.2.4. We compare the three global indexing methods (cumulated histogram, cumulated probability, segment-statistical model) to the frame-based decision method. For the "segment-statistical model" method, we only present the results obtained using the cosine-distance method (using only the mean of the bins c_i) since it leads to the highest results. However, we indicate in Fig. 2 both the mean and standard-deviation of the bins c_i for the 6 classes of our training set.

Music Genre	Classical	Jazz / Blues	World	Electronic	Metal / Punk	Rock / Pop	Total
Training set	320	26	106	115	45	101	713
Development set	320	26	122	114	45	102	729

Table 5: *Description of the training and development corpus for music genre recognition*

		Found					
		classical	electronic	jazz_blues	metal_punk	rock_pop	world
Real	classical	90,6	0,0	0,3	0,0	0,0	9,1
	electronic	1,8	73,7	0,9	2,6	9,6	11,4
	jazz_blues	0,0	0,0	96,2	0,0	3,8	0,0
	metal_punk	0,0	0,0	0,0	84,4	15,6	0,0
	rock_pop	0,0	4,9	2,9	16,7	67,6	7,8
	world	16,4	4,9	4,9	0,8	13,1	59,8

78,7

Table 6: *Confusion matrix of the music genre system for the 6 categories using the 'segment-statistical models".*

The classification on a frame-basis (87039 frames have to be classified) gives a mean Recall of $\overline{R} = 62.2\%$ ($\pm$ 14.3% variation among the classes). The "cumulated histogram" method (729 tracks have to be classified) gives a mean Recall of $\overline{R} = 76.2\%$ ($\pm$

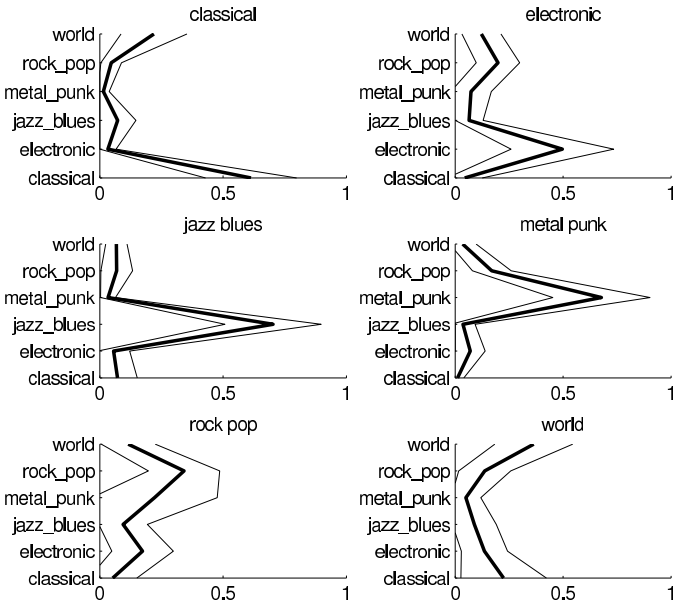

Figure 2: *Trained "segment-statistical models" for the 6 music genres: mean values (thick lines), ± standard deviation (thin lines).*

18.9%). The "cumulated probability" method gives a mean Recall of $\overline{R} = 77.4\%$ ($\pm$ 16.8%) The best method is the "segment-statistical model" with a mean Recall of $\overline{R} = 78.7\%$ ($\pm$ 14%).

In Fig. 6, we indicate the confusion matrix obtained using the "segment-statistical models". The largest confusions occur between classical and world (some world-music tracks use classical instruments), metal-punk and pop-rock (some rock songs are close acoustically to metal songs), electronic and pop-rock. It is difficult to further comment on these confusions considering the spread of the acoustical content of the categories. However, it seems clear that using only timbre-related features (such as the MFCCs and SFMs) do not allow distinguishing high-level concepts such as music genre.

Conclusion: Considering again the fact that no specific modifications of our system have been made for the specific task of music genre recognition, the results obtained are very encouraging. We get a mean Recall of 78.7%. In comparison, the results obtained by the 5 participants of the ISMIR2004 music genre contest[21] were $\overline{R} = 78.78\%, 67.22\%, 58.60\%, 55.70\%, 51.48\%$. It is important to note however that we present results for the development set while the results presented in [21] were for the evaluation set.

4. CONCLUSION AND FUTURE WORKS

In this paper we presented a generic system for audio indexing (classification/ segmentation). The system aims to be easy to use and applicable to a large range of indexing problems. We tested this by applying it to two usual problems: speech/ music segmentation of radio stream (in a real industrial framework) and music genre recognition. For this, a set of generic low-level audio features (MFCC and SFM) was used.

For the speech/ music segmentation problem, and when considering only the pure categories speech/ music, the performances of our system were good. We also showed that the system could be generalizable across datasets: a model trained on a specific radio channel could be used to index other radio channels. However when taking into account the non-pure categories ("mix" and "bed"), the performances of our system dropped.

For the music genre recognition problem, since the indexes to be found are global, we used the proposed "segment-statistical-models" leading to results close to the state of the art.

The main goal of this paper was to show the effectiveness of our generic generic system to solve quickly a specific problem. Considering the fact that we have used the same system for both problems, the results obtained are encouraging.

However the features used were very generic and therefore do not allow to represent precisely the characteristics of some classes. This was the case for the non-pure categories in speech/ music (describing these categories would involve having the possibility to observe separately the various parts of the spectrum). This was also the case for differentiating some music genres (differentiating them would require higher level musical features such as rhythm patterns, chord succession...). Future works will therefore concentrate on extending the set of audio features on which the feature selection is performed.

Another current limitation of our system comes from the unbalancing of training sets (one class is more represented than the others). In fact, real life training sets are often unbalanced. Future works will therefore concentrate in adapting our training algorithms (feature selection, feature space transforms) to this.

5. ACKNOWLEDGEMENTS

Part of this work was conducted in the context of the French RIAM project "Ecoute" (http://projet-ecoute.ircam.fr/) . Many thanks to D. Bachut for the fruitful discussions and D. Tardieu for paper corrections.

6. REFERENCES

[1] E. Frank, L. Trigg, M. Hall, and R. Kirkby, "Weka: Waikato environment for knowledge analysis," 1999-2000.

[2] F. Pachet and A. Zils, "Automatic extraction of music descriptors from acoustic signals," in *Proc. of ISMIR*, Barcelona, Spain, 2004.

[3] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle, "jaudio: A feature extraction library," in *Proc. of ISMIR*, London, UK, 2005.

[4] C. McKay, R. Fiebrink, D. McEnnis, B. Li, and I. Fujinaga, "Ace: A framework for optimizing music classification," in *Proc. of ISMIR*, London, UK, 2005.

[5] G. Tzanetakis and P. Cook, "Marsyas: a framework for audio analysis," *Organised Sound*, vol. 4, no. 3, 1999.

[6] (International Music Information Retrieval Systems Evaluation Laboratory) IMIRSEL, "Introducing m2k and d2k," in *Proc. of ISMIR*, Barcelona, Spain, 2004.

[7] G. Peeters, "Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization," in *Proc. of AES 115th Convention*, New York, USA, 2003.

[8] K. Sjolander and J. Beskow, "Wavesurfer - an open source speech tool," in *ICSLP*, 2000.

[9] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project.," Cuidado i.s.t. report, IRCAM, 2004.

[10] G. Peeters, A. Laburthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. of ISMIR*, Paris, France, 2002, pp. 94–100.

[11] B. Whitman and D. Ellis, "Automatic record reviews," in *Proc. of ISMIR*, Barcelona, Spain, 2004.

[12] M. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. of ISMIR*, Baltimore, US, 2003.

[13] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier, "The ester phase ii evaluation campaign for the rich transcription of french broadcast news," in *European Conf. on Speech Communication and Technology*, 2005.

[14] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. of IEEE ICASSP*, Munich, Germany, 1997.

[15] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proc. of IEEE ICASSP*, 1996.

[16] M. Carey, E. Paris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *Proc. of IEEE ICASSP*, Phoenix, AZ, USA, 1999.

[17] H. Harb and L. Chen, "Robust speech music discrimination using spectrum's first order statistics and neural networks," in *ISPAA*, Paris, France, 2003, pp. 125–128.

[18] J. Pinquier and R. André-Obrecht, "Audio indexing: Primary components retrieval - robust classification in audio documents. dans : Multimedia tools and applications," *Multimedia Tools and Applications*, vol. 30, no. 3, pp. 313–330, 2006.

[19] G. Richard, M. Ramona, and S. Essid, "Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams," in *Proc. of IEEE ICASSP*, Honolulu, Hawaii.

[20] J.-J. Aucouturier and F. Pachet, "Representing musical genre : A state of art," *Journal of New Music Research*, vol. 32, no. 1, 2003.

[21] ISMIR2004, "Audio description contest - genre/ artist id classification and artist similarity," 2004.

[22] MIREX, "Music information retrieval evaluation exchange," 2005, 2006, 2007.

[23] D. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast," in *Proc. of ICME (IEEE Int. Conf. on Multimedia and Expo)*, 2002.

[24] J. Burred and A. Lerch, "A hierarchical approach to automatic musical genre classification," in *Proc. of DAFX*, London, UK, 2003.

[25] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[26] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature sets," in *Proc. of ISMIR*, Barcelona, Spain, 2004.