TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

NATALIA IBÁÑEZ SÁEZ

ACOUSTIC SCENE CLASSIFICATION USING CLUSTERING-BASED ACOUSTIC MODELS

Master of Science thesis

Examiner: Dr. Annamaria Mesaros
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 31st May 2017

ABSTRACT

NATALIA IBÁÑEZ SÁEZ: Acoustic scene classification using clustering-based acoustic models
Tampere University of Technology
Master of Science thesis, 44 pages
May 2017
Master's Degree Programme in Telecommunication Engineering
Examiner: Dr. Annamaria Mesaros
Keywords: Acoustic scene classification, ASC, unsupervised learning, clustering

This thesis consists in the extension of the baseline system for Acoustic Scene Classification, developed by the Audio Research Group at Tampere university of Technology for the challenge of Detection and Classification of Acoustic Scenes and Events (DCASE). The baseline is based on a supervised classification approach which is composed by training and testing stages. The training stage is based on the construction of a statistical model capable to describe each of the environmental classes that will be used during the training stage. The innovation part has the goal of clustering the available observations so that each class is divided into some subclasses. The models will be created for each subclass. These models describe acoustic environments in more detail, which allows achieving higher level of accuracy.

The system has preserved its previous stages and the method used for the clustering has been k-means. The experiments have been performed firstly with the development dataset and the results obtained have been validated with the challenge dataset aiming to verify that the system is capable to generalize its results. Three different approaches have been tested: First, the number of clusters has been set invariant for all the classes. Values 2, 3, 5 and 10 have been tested. The performance has increased 2% for 2 clusters. Second, the number of clusters has been selected manually choosing the values that proved to provide better performance for each class during the development stage. The performance has increased 2.3% with respect to the baseline. Third approach is more sophisticated and includes cluster evaluation based on BD and CH indices. This method allows calculating the number of clusters for each class automatically. It has improved the performance in 2% with respect to the baseline.

PREFACE

This thesis is dedicated to Tampere University of Technology and all the Finnish people that I have met, for giving a perfect year here and making me feel so at home.

To Annamaria, for letting me being part of this project, for her invaluable help, for her infinite patience and, in conclusion, for making so great working in this project.

To Nuria for her contributions to this project.

And specially, to all the friends of my Erasmus that have made that this year has been like a dream.

Y a mamá, papá y Blanca, que siempre se preocupan de que sea feliz.

Tampere, 24.5.2017

Natalia Ibáñez Sáez

TABLE OF CONTENTS

1. Introduction 1
1.1 Motivation 1
1.2 Objectives 2
1.3 Methodology 3
1.4 Tools 4
1.5 Structure 4
2. State of the Art 6
2.1 Overview of Acoustic Scene Classification 6
2.2 Related work and applications 6
2.3 Acoustic Scene Classification system functionality 7
2.3.1 Feature extraction 9
2.3.2 Clustering 11
2.3.3 Classification methods 14
3. Design 16
3.1 Introduction 16
3.2 System overview 18
3.3 Cross-validation setup 18
3.4 MFCC features 19
3.5 Feature normalization 22
3.6 Clustering 22
3.7 GMM statistical model 25
3.8 Evaluation 28
4. Experiments and results 29
4.1 Database 29
4.2 Parameters 29

4.3 Results 30
4.3.1 Development dataset 31
4.3.2 Challenge dataset 33
5. Conclusions 41
Bibliography 43

LIST OF FIGURES

2.1 Block diagram of ASC systems. 8

3.1 Description of ASC problem. 16

3.2 Cross-validation setup. 19

3.3 Block diagram of MFCC features. 20

4.1 Comparison of performance between the baseline and the innovation system for the development data. 33

4.2 Comparison of performance between the baseline and the innovation system for the challenge data. 38

4.3 Comparison of performance between all the methods for the challenge data. 40

LIST OF TABLES

4.1 Performance for fixed number of clusters for the development data. . 31

4.2 Performance for the optimal number of clusters selected manually for the development data. 32

4.3 Number of clusters selected by CH index for the development data. . 34

4.4 Number of clusters selected by DB index for the development data. . 35

4.5 Performance for the number of clusters selected by CH index for the development data. 36

4.6 Performance for the number of clusters selected by DB index for the development data. 36

4.7 Performance for fixed number of clusters for the challenge data. . . . 37

4.8 Performance for the optimal number of clusters for the challenge data. 37

4.9 Performance for the number of clusters selected by CH index for the challenge data. 38

4.10 Performance for the number of clusters selected by DB index for the challenge data. 39

4.11 Comparison of performance between all the methods for the challenge data. 39

LIST OF ABBREVIATIONS AND SYMBOLS

ASC	Acoustic Scene Classification
CASA	Auditory Scene Analysis
DCASE	Detection and Classification of Acoustic Scenes and Events
MFCC	Mel Frequency Cepstral Coefficients
GMM	Gaussian Mixture Models
ITD	Interaural Time Difference
ILD	Interaural Level Difference
LPC	Linear Predictive Coefficients
PCA	Principal Component Analysis
ICA	Independent Component Analysis
DB	Davies-Bouldin
CH	Calinski-Harabasz
HMM	Hidden Markov Models
SVM	Support Vector Machine
k-NN	k-Nearest Neighbour
ML	Maximum Likelihood
MAP	Maximum a Posteriori
DFT	Discrete Fourier Transform
DCT	Discrete Cosine Transform
PDF	Probability Density Function
EM	Expectation-Maximization

1. INTRODUCTION

1.1 Motivation

In recent times, many research fields in technology have been trying to implement methods that can perform the task that are carried out by humans. This is the purpose of machine learning; whose main purpose consist of trying to build machines capable of accomplishing task that humans can done in an innate way.

Although humans can perform some task in an innate manner, that does not require extra effort and can be carried in an intuitive manner, replicating these capabilities in a machine represents an important challenge, which is not a trivial matter. In that scenario, machine learning becomes a field of great importance nowadays.

Machine learning is the subfield of computational sciences and a branch of artificial intelligence, whose purpose is developing machines that permit computers learn how to behave based on some rules trying to replicate the intelligence of humans.

In this context, one of the capabilities that can be desired that a machine performs are those related to human senses. Very well-known are the advantages that artificial vision is bringing to our society, encompassing several applications related to the human vision, as described in [1]. Likewise, the field of Computational Auditory Scene Analysis is currently seeing very active development [2]. In that sense, this project aims to approach and contributing to the development of the studies audio analysis for machine learning.

One of the specific problems studied in audio analysis for machine learning is Acoustic Scene Classification (ASC), which aims to characterize a sample of audio using a label that explains where the audio is recorded. The classification of different environments is made based on the sound that are hallmarks of each environment.

ASC is a subproblem comprised in a more general field named field named Com-

putational Auditory Scene Analysis (CASA), which has been studied in different research problems. As described in [3] some of this research includes development of methods for the classification of noise sources, algorithms for sound source recognition [4], and identification and labelling temporal regions where single events that belongs to specific classes are present.

The scope of applications that CASA can provide is very varied. Some of the most popular includes surveillance systems [5], elderly assistance [6], context-aware services [7], intelligent wearable devices [8], robotics navigation systems [9] or and audio archive management [10].

1.2 Objectives

The main purpose of this projects consist in contributing to the study of ASC. In that way, this project is intended to extend an existing classification system and if possible improve its performance.

The baseline system consists of a tool developed by the Audio Research Group at Tampere university of Technology, implemented as a baseline system for the challenge on Detection and Classification of Acoustic Scenes and Events (DCASE). DCASE is a project created by this university in an effort to o support interest in this research area and provide the research community with a starting point for data collection and common evaluation procedure [11]. This project aims to extend the DCASE baseline using an existing database provided by the Audio Research Group at Tampere university of Technology.

The baseline which in represents the starting point of this project, is based on two main pillars: the extraction of audio characteristics using Mel Frequency Cepstral Coefficients (MFCC) and the supervised classification of audio using Gaussian Mixture Models (GMM). The objective of this project is to investigate if it is possible to achieve better classification performance by introducing an unsupervised clustering block between the MFCC extraction and GMM supervised classification.

The role of the unsupervised clustering is to split each labelled class into different subclasses that provides a better description of the environment.For example, if we consider the class labelled as “beach”, its subclasses can be “windy beach”, “silent beach” or “crowded beach”. The point of this subclassification is that the

GMM modeling each subclass gives a more accurate model that can permit a better generalisation for classification of unseen audio examples.

The other objective consists in analysing this results to study quantitatively the influence of the unsupervised clustering in Audio Scene Classification.

1.3 Methodology

To carry out the study of the influence of introducing unsupervised clustering in the existing software baseline, the methodology that will be followed is presented below:

First of all, a bibliography review will be performed. Some of the history and alternatives for ASC will be presented and the positive and negative aspects about all of them will be presented. The review will show the different approaches and methods that have been used through the history of the ASC research. In addition, a study of unsupervised machine learning will be performed. The clustering methods will be presented in more detail since they are crucial in this project. The innovation part of the project is based on unsupervised clustering and therefore, special attention will be paid to clustering methods.

Later, each steps of the baseline system will be presented, so that it become clear to understand how the baseline stages and the innovation part are integrated. The innovation part of the project will be situated right before the model training stage so that the mathematical models can be constructed based on the new subclasses instead of on the original classes. This part will consist on an unsupervised clustering that will split the bag of features asociated to each class into some bag of features asociated each of them to a subclass. After that the pipeline will perform the same tasks as before but constructing the models based on the subclasses. The classification in the testing stage will now be slightly modified so that the system now will decide the class when one of its subclasses is the model where it best fits the test audio file.

Finally, an analysis in terms of the performance of the system will be done. Some parameters will be tuned in order to optimize system performance. The parameters that will be tuned will be especially those concerning the clustering stage, but also some other parameters that belong to other stages will be modified since they are affected by the changes in the previous stages. The obtained data will provide a

basis to quantitatively decide what is the influence of the newly introduced clustering stage to the functionality of the system.

1.4 Tools

This project will be developed using Matlab. The baseline used is also programmed in this language, and is developed by TUT [12]. It includes some external libraries that are Rastamat for feature extraction and Voicebox for GMM models.

It will also TUT Acoustic Scenes 2016 dataset [12] that contains several audio files that will be analysed to measure the functionality of the developed system.

1.5 Structure

This thesis is structured as follows:

Chapter 2 presents a literature review concerning ASC. It presents the different approaches and methods that have been used to implement ASC available in scientific literature. It also mentions different applications of ASC systems. Finally, it explains the techniques than can be used for ASC.

Chapter 3 is dedicated to describing the implementation of the system, including both the baseline part and the innovation part. It shows the block diagram of the system and after that, it explains carefully each block of the pipeline, showing the functionality of this part in its specific implementation. It also clarifies in which point of the pipeline the innovation is introduced. It includes both the explanation of the algorithms implemented as well as the design decision taken, such as the parameters to be tuned for the study.

Chapter 4 presents the results obtained and analyses them in order to calculate the effect of the clustering step to the overall system. In particular, it compares the performance reached by the baseline and the extended system to justify the importance of including an unsupervised clustering stage in order to improve the representation obtained by the constructed statistical model, and, in consequence, the accuracy of the system.

Finally, Chapter 5 presents the conclusions obtained based on this experiment and

gives some ideas on how to continue the study of unsupervised clustering in ASC research.

2. STATE OF THE ART

2.1 Overview of Acoustic Scene Classification

Acoustic Scene Classification can be defined as the task of characterizing a sample of audio using a label that explains where the audio is recorded. The task consists in processing some labelled audio files to construct an statistical model that can be generalized for future and unlabelled audio files.

The most common way to perform this task consists in extracting characteristic features that can be used to classify new audio files. Acoustic signals contain a lot of redundant information. Audio features features serve as quantitative way to summarize the most important information in an audio file. They are capable of reducing this redundant information and creating a compact representation. Audio features help to retain smaller amount of the acoustic information present in the audio files.

The audio features will be used to construct a model. A statistical model is the crucial stage of supervised learning. The model is the tool that makes possible describing the currently labelled segment files so that the system has a general knowledge on how the acoustic environment is. This model should have good generalization properties. It means that when unlabelled audio segments are fed to the system, the system outputs the label of the environment in which the audio segment is most likely to have been recorded.

2.2 Related work and applications

There are some other fields of audio processing that are similar to ASC. Some of them are the following: classification of noise sources, that can be used, to carry out tasks as improving the performance of speech processing algorithms; sound source

recognition, in which the purpose is identifying the sources of the acoustic events by naming the object that produces the sound.

One related field that deserves special mention is sound event detection. Sound event detection is defined as recognition of individual sound events in audio [12]. This field is defined in the same way that ASC from pattern classification point of view, because they both have predefined classes and constructing models. Sound event detection is therefore supervised classification. They are related since ASC sometimes benefits from event detection since in some situations the identification of events in an audio file can characterize the general environment [3]. An example of an application of sound event detection is semantic analysis of audio streams that can be used in automatic indexing and retrieval systems.

The range of applications for ASC is very wide, and there are many interesting fields where it can be useful. To name the most important, context-aware services [7], intelligent wearable devices [8], robotic navigation system [9] and audio archive management [10] can be included.

Smartphones can also benefit from using of ASC. One example can be developing software for smartphones capable of becoming aware of their acoustic environment, which permits them to switch between sound or silent mode depending on the necessities of its acoustic context.

Assistive technologies is another field that can benefit from the developing of ASC. As sake of example, wheelchairs that can switch their functionalities depending on if they are indoor or outdoor. Also, indexing and information retrieval in the field of audio applications benefits from this research. These systems can extract content information related to the audio files that can be converted into labels for these files.

2.3 Acoustic Scene Classification system functionality

In most of the systems for ASC the pipeline is similar, and consists of two main branches: the training line and testing line. Figure 2.1 represents a typical block diagram for ASC systems.

It can be observed that the system it formed by two branches. The upper branch is dedicated to the training stage, which aims to create the models for the acoustic environments. The lower branch, is dedicated to the testing stage. It uses the

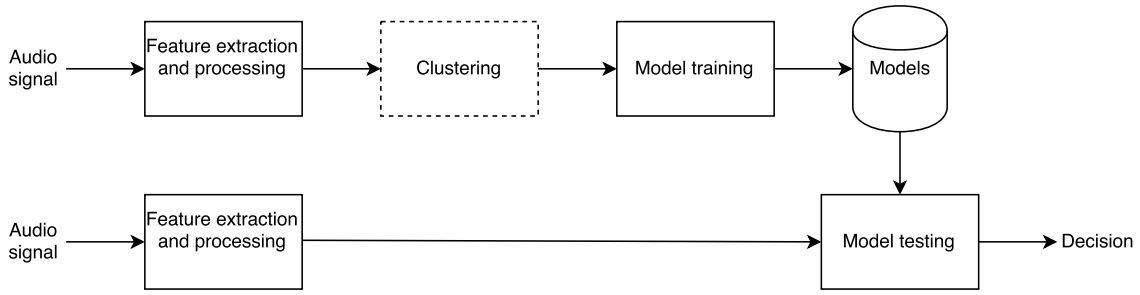

Figure 2.1 *Block diagram of ASC systems. The system classifies audio examples given as input into one of predefined classes. The blocks of the baseline are represented in solid line and the innovation block is represented in dashed line.*

models created during the training stage to provide the decision of the class which is the output of the system.

The pipeline is composed by the following blocks:

Both branches begin with a feature extraction and processing step. Its purpose is keeping the information that is relevant in some way due to its acoustic characteristics. This block is very important because the calculated acoustic features perform the comparison between audio files. The features have to be postprocessed. Some of the most usual postprocessing of the acoustic features include normalization such that the resulting feature set has zero mean and unit variance.

The clustering block is the functionality that corresponds to the innovation of this project. Is not part of the original baseline and for that reason it will be explained at the end of this section.

The model training block is a crucial block in this ASC systems. Its task consist in creating a model that is capable to represent the characteristic properties of each environmental acoustic class. Each class is represented by a different model. The models are created based on the features extracted from the audio signals. The way that the model is created is in a supervised machine learning approach, since correspondence between each class and the audio files belonging to it is provided to the system.

Regarding to the training branch, the audio file is analyzed the same way as for training, by extracting the same acoustic features and doing same postprocessing.The model testing block is in charge to give a decision when a new audio signal is fed

to the system. Its task consists in determining for unlabelled files in which of the classes they fit better and therefore output the label of the environment where the audio is supposed to have been recorded.

The clustering block is an unsupervised machine learning stage, whose purpose is dividing each class in some subclasses, with some aspects in common that differentiates each of them from the others. As unsupervised learning method, no prior knowledge is available to achieve this subclassification so its purpose is creating certain subclasses that are hidden in the observation.

2.3.1 Feature extraction

Feature extraction is a crucial process in the development of the system. The features bring the possibility to make comparisons between files paying attention to the most relevant characteristics of the audio. It also helps to make lighter the following processing steps. The original audio information was very large and after the features are extracted, a relative small number of coefficients will be enough to represent the main acoustic information in the audio.

A summary of feature descriptors used for ASC is presented in [3], and summarized in the following:

- Low-level time-based and frequency-based audio descriptors: This type of features are the easiest ones that can be calculated. It can be calculated from the signal domain or time, such as the crossing zero rate and from the frequency domain, such as the center of mass of the spectrum and the spectral roll off, which determines the frequency where the magnitude of the spectrum falls below a threshold.
- Frequency-band energy features (energy/frequency): This feature measures the amount of energy contained in each frequency band by integrating the energy of each band. It can also be calculated as ratios of each energy in comparison with the total energy to keep the frequency regions of the signal with more energy.
- Auditory filter banks: This features encodes the energy of the frequencies according to a frequency scale that is inspired by the human auditory system.

Examples of these features are the Gammatone filters, the Mel-scaled filterbank coefficients (MFCs) and features based on the auditory spectrogram.

- Cepstral features: These features are based on MFCs features but they also perform the logarithm and DCT transform. They capture the spectral envelop of sounds, which helps to summarize the spectral content within less coefficients. MFCCs are the most popular example of them and are very commonly used in ASC.

- Spatial features: These features can be obtained when multiple microphones have been used. The spatial features help to capture the properties of the acoustic scene. The measures that can be used are the interaural time difference (ITD) that has information about the difference between left and right channels and the interaural level difference (ILD), that measures the difference between the amplitude of the channels.

- Voicing features: These features are valid for the signals that include harmonic components. In this case, some specific events can be modelled based on their harmonic structure and it can help to identify more easily the acoustic scene. The cochleogram is a representation inspired in the human clochea and its representation helps to identify tonal events in acoustic scenes.

- Linear predictive coefficients (LPCs): This features are based in the idea of autoregressive models, that represent a signal at a given instant as a linear combination of samples at the previous instants. The LPC and the spectral enveloped are directly related so these features encode the spectral information of a sound.

- Parametric approximation features: This feature is a generalization of the previous one. In this case, the value of the signal can be parametrized by a set of parameters. These methods include for example the use of Gabor filters.

- Unsupervised learning features: The features can be learned with no prior knowledge of the properties. This knowledge will serve to form the bases functions than can characterize the acoustic signal. One typical method for that is a sparse restricted Boltzmann machine (SRBM) that is a neural network approach.

- Matrix factorization methods: These features are based in the idea of decomposing the spectrogram as a combination of elementary functions. The

important information therefore, will be included in these functions and the feature will be calculated based on these functions. It is an unsupervised learning method because the functions are not known in advance.

- Image processing features: These features are inspired in image processing techniques. An example of them is a technique that based on the constant-Q transform of the audio signal, it creates images by interpolating neighbouring time-frequency bins. This feature is finally obtained calculating the histogram of the local gradients.

After the features are extracted, they can also be processed in order to improve the performance of the system.

Some common methods are described in [3] and they are summarized below:

- Feature transforms: The purpose of this type of processing is enhancing the extracted features. One of the most common methods is applying dimensionality reduction methods. This methods get to reduce the components of the features, which bring generalization properties. Then, redundant information present in a feature can be eliminated, and only the most important and characteristic information is kept.

  Some approaches are principal component analysis (PCA) and independent component analysis (ICA). Both methods have in common that they project high dimensionality data into lower dimensionality following the criterion of maximizing the variance of the data.

- Time derivatives: The idea of time derivatives is capture the dynamic information of the audio signals. They are produced deriving consecutive audio feautures and they are included after the original coefficients

2.3.2 Clustering

Clustering is a method comprised in the field of unsupervised machine learning. The objective of unsupervised machine learning is building a model that describes a set of observations when no prior knowledge of the nature of them is available. The system is not fed with any information about the observations and thus, the

specific characteristics of each group underlaying within the class is self-learned by the machine.

Clustering is a specific method of unsupervised machine learning. Its objective is dividing the observations into some groups whose components have similar characteristics among them and different to the components belonging to the rest of the groups. In this particular system, the clustering stage has the goal of dividing each class into some subclasses which permits building a model that represents more accurately the observed data. Different clustering methods are presented below [13]:

Hierarchical clustering: The dataset is partitioned by levels so that in each level generally two groups of the previous levels are connected or divided, depending on if they are agglomerative algorithms or divisive algorithms, respectively.

- Single Link: In each step the two groups whose elements have the minimum distance are joined.
- Average Link: In each step the two groups whose elements have the minimum average distance are joined.
- Complete Link: In each step the two groups whose diameter is minimum or whose maximum within distance is minimum are joined.

Partitional clustering: These algorithms make an initial division of the data in groups and moving afterwards the objects from one group to another aiming to optimize a certain function. These algorithms require a priori the number of clusters in which are going to be distributed. Some of these algorithms are the followings:

- K-means: This algorithm has the purpose of dividing the data into a given number of clusters that contains each of them a centroid. To do that, initial centroids are defined and the data are grouped around to the centroid to which they are closer. After that, the centroid of each cluster is recalculated and all the data are redistributed based on to which centroid they are closer. This process is repeated until the convergence is reached.

- CURE: This algorithm is a hybrid between partitional clustering and hierarchical clustering approaches that tries to solve the disadvantages of each of them. The idea is that for each group more than one point is selected. These points are calculated. These points are calculated based on the most disperse points of the group, which are moved towards the centre with the same compression factor and in each steps the closest points are connected and once they are connected the most representative points are recalculated.

Density-based clustering: These algorithms are based in the idea of dividing the closest elements of a basis in groups considering the density distribution of points, with the objective that the groups that are formed have a high density of within points whereas the density between them is low. These algorithms use diverse techniques such as the graph theory, histogram based techniques. They use the concept of central point, edge or noise.

The clustering methods mentioned above have in common the fact that the number of clusters must be decided in advance and be given as a parameter to the clustering algorithm. This leads to the disadvantage of that the output clusters are not guaranteed to be optimal provided that the underlying number of classes can be different from the given number of classes and therefore, the clustering does not provide a good description of the underlying partitions. To solve this problem measuring of the degree of performance can be calculated for several number of partitions. Based on the results, the number of partitions that best describes the underlying groups can be selected and the clustering can be performed fixing this optimal number as the number of clusters that will be calculated.

The level of accuracy of a clustering, usually considers two different measures: the within distance, which measures the distances between the observations that correspond to the same cluster and the between distance, which measures the distance between the different clusters. The within distance is desired to be minimized whereas the between distance is intended to be maximized. The function used to minimize and maximize these distances is different for the different evaluation methods. Some of the most common methods for measuring the quality of the resulting clusters are Davies-Bouldin (DB) index, Calinsk-Harabasz (CH) index, and silhouette criterion. All of them have in common that they are based on the idea of maximizing the distances between the different clusters and minimizing the distance of the points belonging to the same cluster. The difference between them is the specific function

that relates those terms and which they try to optimize.

2.3.3 Classification methods

In a classification system, statistical models have the role of describing the features of the classes so that the model can be generalized and used to classify unlabeled unlabelled data.

The training stage consists in calculating a statistical model that is capable to represent the acoustic properties of each subclass with the objective of generalizing this model to new unlabelled audio segments. This new audio segments will be classified during the training stage.

The statistical model is calculated based on the features extracted in the previous stage. It is important to note that the method used is developed in a bag-of-features approach, which means that temporal information is not considered. The model for each subclass can be viewed as a bag where all the features are introduced and they are considered as a basis for constructing the model.

A summary of statistical models used for ASC is presented in [3], and summarized in the following :

- Descriptive statistics: This method is capable to describe characteristic of the statistical distribution of the features, including mean, variance, skewness, kurtosis of a distribution, quartiles or percentiles.
- GMM: This method describes a model by the linear combination of Gaussian basis. The number of Gaussian basis is given to the system and the task of the system is determining the parameters of the multivariate Gaussians. The obtained function describes statistically the observed data.
- Hidden Markov Models (HMMs): This method can be viewed as a generalization of GMM. It includes the distribution function that best generates the features observed but it also takes into account the temporality of the scene. It includes matrixes with information about the probability of transition between different events so that they can analyse complex soundscapes
- i-vector: This models are obtained based on a sequence of GMM functions.

The GMM are calculated based on MFCC features. This method is specially used in speech processing, because it permits verifying a speaker.

The training stage aims to classify the unlabelled audio files into one of the predefined classes based on the level of coincidence to the model for each class. The most common decision criteria are listed below [3]:

- One versus one and one versus all: This decision criteria is intended to be used with a model based on Support Vector Machine (SVM) approach. In this case, the position of a feature vector is mapped to a class.
- Majority vote: This decision criteria refers to the fact the system takes its decision based on different moments of the segment. The output can be decided based on the most common category or can be decided weighting differently each moment.
- Nearest neighbour: In this case the output is the class whose feature descriptor has the shortest distance to the feature descriptor of the incoming signal. It can also be generalized to k-nearest neighbour (kNN), where the decided class is that to whom the majority of the input feature descriptor is shortest.
- Maximum likelihood (ML): This criterion is used when generative models are constructed, such as GMM and the class is selected based on which class model is more likely to have produced the observed data.
- Maximum a posteriori (MAP): This criterion is a generalization of maximum likelihood. It includes information about a priori distributions so that by the Bayes theorem, the probabilities a posteriori can be calculated to construct models with more information.

3. DESIGN

3.1 Introduction

This chapter explains the choices for each block of the classification system that serves as a baseline for development that are used through the system pipeline that serves to classify audio files into given classes based on the audio content by performing supervised and unsupervised classification as depicted in figure 3.1 [12].

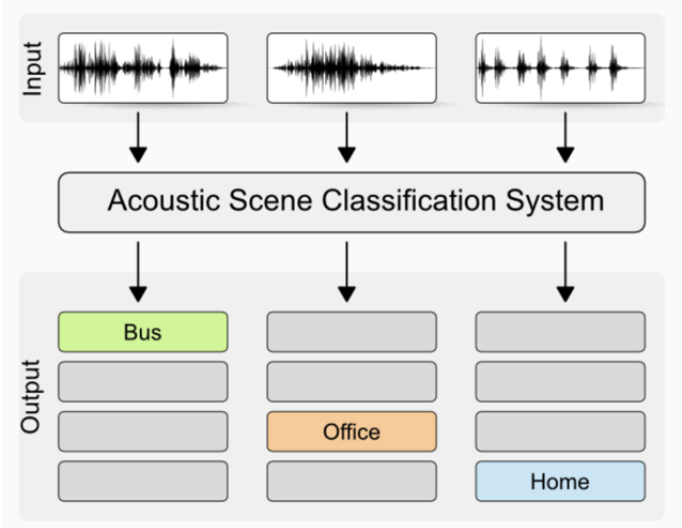

Figure 3.1 *Description of ASC problem. ASC systems classify audio file inputs and give as output the name of the class to which they belong.*

The system developed for this project is based on the idea of extending the available baseline of DCASE 2016 [12]. This section includes description of the techniques used in the baseline system, and in addition, the unseupervised learning method chosen for the extension of the baseline. In this project the techniques used are the ones included in DCASE baseline system [12] with the inclusion of clustering for the unsupervised learning.

The extracted features are MFCC since they have shown to be efficient for achieving similar performance that the humans achieve. This happens because these features consider the information of the acoustic human system. Therefore, the errors that occur in the system are very similar to the ones produce by humans.

The models used are GMM for supervised classification, since they give good performance for the description of acoustic scenes and the decision criteria for the supervised learning stage is maximum likelihood.

The unsupervised clustering will be performed using clustering based on k-means algorithm and evaluation techniques are used with the purpose of clustering the features of a class in the optimal way aiming to be able to construct models that describe the data of a subclass as best as possible. Clustering is a technique that belongs to the area of unsupervised learning. This technique is capable to produce a division of the data when no prior knowledge is available. The clustering block is in charge of determining the patterns of the data in terms of hidden groups of data that share some characteristics that makes them be more similar between them and more different to the others.

K-means is an algorithm that performs clustering of observations with no prior knowledge of the nature of the data, however it requires as information the number of clusters that need to be created. The number of clusters is a parameter that must be chosen, even if there is no logical evidence that determines the number of clusters in which all class should be divided. Moreover, the number of underlying groups inside a class could be different for each class. This is a challenge for the system, that can be solved in different ways. In this particular project, this challenge will be solved in two different ways.

The first way is a simple approach and consist in proceeding in a more manual way. The number of clusters can be chosen based on the performance obtained during system development. However, for some classes theperformance obtained using clustering may be lower because for them the selected number of classes may not be optimal. An automatic way consists in determining dynamically the number of clusters that best describes the distribution of the components of each class. This approach considers that for each class the number of clusters can be different and the number of clusters can also vary for different folds of the same class. In order to perform this optimization, methods that evaluate the quality of clustering of the clusters must be included into the pipeline.

In the following sections, the stages that are part of the system, will be described in more detail. The description will include both, the stages of the baseline system as well as the methods includes in the unsupervised clustering.

3.2 System overview

The objective of the system consists basically in taking audio files as an input and giving the name of the acoustic environment that they belong to as an output. The classes are known in advance, and the system is provided with a dataset consisting of several audio files recorded in each of the environment classes. The system therefore, needs samples of audio for each environment class to become capable of classifying the audio input. This is called supervised learning, because the system is learning to recognize patterns of new observations based on previously seen observations that were labeled by humans.

The unsupervised learning, on the contrary, refers to the scenario where the information that is given as input to the system is not labelled and the machine is the one that must learn the patters with no external help. In this context, the system will oversee the task of determining the different underlying groups that exist in an acoustic environment.

The proposed system integrates both types of machine learning and the system has been designed according to the stages described below.

3.3 Cross-validation setup

The experiments will be executed in a cross-validation way. This means that the audio files will be separated into different subsets as training and testing data, that effectively repeats the experiment as if the number of files were four times more. This method is very convenient to be used when the amount of data available is not large enough or if more data is wanted to give more accurate results.

The available data for the development stage is divided so that the experiment is executed four times. To provide four set of experiments the data will be divided as depicted in the figure 3.2. It can be observed there that the developing data had been divided in four different ways. In each of the four experiments, three quarters

of the data are considered as training data and the rest as testing data. The subset of data that are taken as testing data are distinct.

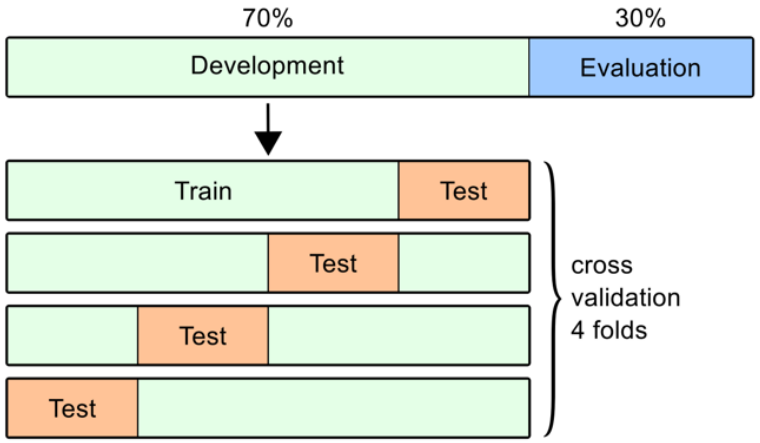

Figure 3.2 *Cross-validation setup provided with baseline system (figure taken from DCASE2016 website). The development data is divided into 4 validation folds non-overlaping.*

It is also important to take into account when making the division between training and testing data that the audio segments that correspond to the same audio recording have to be all included either in training or testing subset. The reason is that if they are included in both subsets it leads to overestimating system performance because the models will be constructed with recordings very similar to those that will be tested.

In addition, a separate set of segments that will be used for evaluation of the method are not used during the developing process. This will give more credibility to the results. This will prove if the model is well constructed or it was overestimating.

3.4 MFCC features

Mel frequency cepstral coefficients are the most popular features used in ASC systems. They provide information about the spectral envelope of the sounds by summarizing the acoustic information taking into account characteristics about the human auditory system.

The MFCC features are calculated according to the block diagram in figure 3.3.

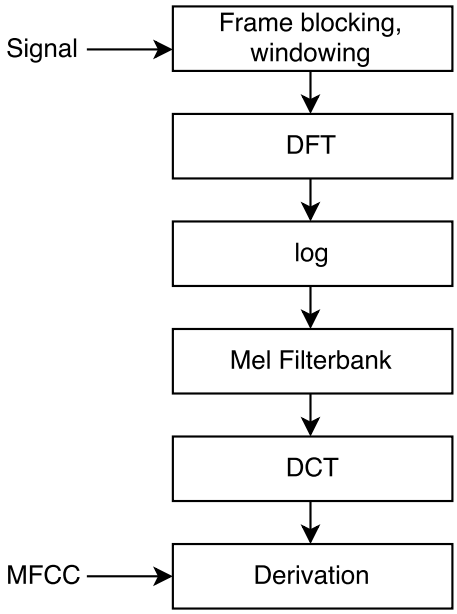

Figure 3.3 *Block diagram of MFCC features.*

As it can be observed in the diagram, the first step consists in segmenting the audio to be analyzed into short time frames in which the features will be calculated. After this point, a feature will be calculated for each of these audio frames. The length of the frames must be chosen enough short that the signal can be considered quasistationary and enough long that there are enough samples to calculate the spectrum of the signal by the Fourier transform in the following blocks. In frame blocking, different windows can be used, such as the Hamming window, which is used to obtain smoothing in the frequency domain and avoid side lobes.

The next step consists in calculating the spectrum of the signal. The motivation for this step is that spectral information was found to be a good way of characterizing audio signals.

To extract the information about the frequency domain of the signals, the DFT is implemented as in equation 3.1.

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N} \quad 1 \leq k \leq K \tag{3.1}$$

where $s_i(k)$ is the framed signal, N is the sample length analysis window, $h(n)$ is

the window function and K is the length of the DFT.

The power spectrum is calculated by squaring the previous signal.

After DFT calculation, only the lower half of the obtained spectrum is further used in processing because the spectrum is symmetrical and therefore keeping all the spectrum will be redundant.

The mel-filterbank block has psychoacoustic motivations. The human auditory frequency perception is not linear and the auditory system is more capable to distinguish between close frequencies if they are situated in the lower frequencies rather than in the highest ones. The aim of this block is mapping the linear frequency scale into a scale that is closer to the human perception scale. The number of mel filters can be selected, but most often 40 filters are used.

The scale in which the acoustic information is percieved by the human auditory system is approximately linear up until 1000 Hz and approximately logarithmic afterwards. The equation that relates linear frequency scale and mel frequency scale are 3.2 for direct transform into mel escale and 3.3 for inverse transform.

$$m(f) = 1125 \ln(1 + f/700) \tag{3.2}$$

$$f(m) = 700(exp(m/1125) - 1) \tag{3.3}$$

The following step is taking the logarithm of the resulting signal. This step is also inspired by the auditory human system. The loudness perceived by humans does not follow a linear scale. The loudness perception follows a logarithmic scale and this is the motivation for calculating the logarithm.

The next step consists on calculating the discrete cosine transform (DCT). For cepstrum calculation, normally inverse FFT is used, but DCT can be used here because the two transforms are equivalent for real data with even symmetry. DCT is capable to summarize the information of the signal in few coefficients as it concentrates most of the energy of the signal in few coefficients. Moreover, DCT results in features that are not correlated, and this property is used in the statistical models implemented in this system. The fact that the MFCC are not correlated allows use of diagonal covariance matrices in the GMM stage, and that simplifies the training stage. The number of coefficients kept is another setting parameter of the system. Usually

in speech recognition 12-13 coefficients are kept, while in other audio classification tasks the number of coefficients used can be higher. In this case we select to keep 20. The first coefficient represents the energy of the signal and is often discarded, however, in this application we select to keep it.

The last steps consist of calculating the delta and delta-delta coefficients. These coefficients are the derivatives of the coefficients obtained previously. Delta coefficients are the first order derivative and the delta-delta are the coefficients are the second order derivatives of the static coefficients. They are calculated according to the equation 3.4.

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{3.4}$$

The complete MFCC feature vector is finally obtained concatenating the original coefficients, the delta coefficients and the delta-delta coefficients.

3.5 Feature normalization

After features are extracted they have to be normalized. The reason is that the features calculated depend on the particular conditions of the situation where the audio was recorded. These particular conditions are not related to the environment class itself but to some other aspects of the moment of the acquisition of the signal, for example the loudness of the sounds that were present in this moment. The purpose of normalization is avoiding the absolute values of the features from having too much influence on cost functions used in the training of the machine learning methods.

During the feature normalization step, the mean of all the features extracted for all the classes is subtracted from the features, and they are divided by its global standard deviation. The resulting features have zero mean and unit variance.

3.6 Clustering

The method used for clustering in this thesis is k-means. K-means is a clustering algorithm belonging to the group of partitional methods. Its purpose is that given

the desired number of partitions of the space, a set of D partitions is divided in K groups that optimize the criterion of the partition. This criterion is minimizing the weighted average of its centroids. The algorithm is performed in 4 steps that are described below:

Fist of all, the number of groups that are going to be formed is decided. Based on this number, K observations are chosen randomly. The groups are at this point only formed by a single observations, which is also the centroid of the group. Thus, at the moment, only K observations belong to K different groups, and the others are not grouped yet.

During the second step, each observation is allocated to the group to which it is the closest based on a measure of distance from the observation to the group centroids. The distance can be measured with different metrics and most often the distance is squared Euclidean distance. Euclidean distance is very influenced by large values and this is one reason for which feature normalization is useful. Zero mean and unit variance features are appropriate for the system.

During the third step, once each object has been assigned to one of the groups, the centroid for each of the K groups is recalculated.

Finally, steps 2 and 3 are repeated iteratively until there are no more reallocations, in other words, until the convergence is reached. Nevertheless, the output obtained is not guaranteed to be optimal because the observations that serve as the first centroids are selected randomly and therefore, the solution highly depends in how these centroids have been selected. To overcome this problem improve the solution, the algorithm is repeated many times such that the initial centroids are different from those of the previous replicates. The output is selected among the replicates by chosing the solution with smallest within distance, which is the average distance from each point.

A drawback of using thismethod for clustering acoustic scenes is that the number of clusters has to be given to k-means algorithm, however, the user does not know in advance how many subclasses there are for each class. If the number of clusters into whom the data are split is not the number of different subclasses that actually exist for each class, then the clusters do not describe a charactersitic subclass. Some approaches to overcome this problem can measure the quality of the resulting clusters. A more rudimentary approach is manual selection of the number of clusters

by tuning this parameter and selecting for each class the number of clusters that provides higher performance for the system.

The methods for evaluating clustering solutions that are used in this thesis are Calinski-Harabasz criterion and Davies-Bouldin criterion. They are two of the best well known methods for evaluation of clustering quality, and they are both available in "Statistics and Machine Learning" toolbox of Matlab.

The **Calinski-Harabasz** [14] criterion is defined by equation 3.5:

$$CH = \frac{SS_B}{SS_W} . \frac{N-k}{K-1} \tag{3.5}$$

where SS_B is the overall between-cluster variance, SS_W the overall within-cluster variance, k the number of clusters, and N the number of observations.

The between-cluster variance is defined as in equation 3.6 and it measures how far are the clusters from each other:

$$SS_B = \sum_{i=1}^{k} n_i ||m_i - m||^2 \tag{3.6}$$

where k is the number of clusters, m_i is the centroid of cluster i, and m is the overall mean of the sample data.

The within-cluster variance is defined as in equation 3.7 and it measures how far are the components of each cluster to each other:

$$SS_W = \sum_{i=1}^{k} \sum_{x \in c_i} = ||x - m_i||^2 \tag{3.7}$$

where k is the number of clusters, x is a data point, c_i is the ith cluster, and m_i is the centroid of cluster i. The optimal number of clusters is that whose CH has the highest value.

The **Davies-Bouldin** [15] criterion is defined by equation 3.9:

$$DB = \frac{1}{K}\sum_{i=1}^{k} max_{j\neq i}\{D_{i,j}\} \tag{3.8}$$

where $D_{i,j}$ is the within-to-between cluster distance ratio for the ith and jth clusters and it can be expressed as:

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}} \tag{3.9}$$

where $\bar{d}_i$ is the average distance between each point in the ith cluster and the centroid of the ith cluster, $\bar{d}_j$ the average distance between each point in the jth cluster and the centroid of the jth cluster, $d_{i,j}$ the Euclidean distance between the centroids of the ith and jth clusters. $D_{i,j}$ is the worst-case within-to-between cluster ratio for cluster i. The optimal number of clusters is is that whose DB has the highest value.

3.7 GMM statistical model

The statistical model used in this project is Gaussian Mixture Model (GMM), which is a parametric model created as weighted combination of Gaussian basis. The idea is to create a probability density function (PDF) that can represent as accurately as possible the distribution of the available features. The equation that describes this model can be seen in equation 3.10 [16]:

$$G(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \tag{3.10}$$

Each component is a multivariate D-dimensional Gaussian function as described in equation 3.11.

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\{-\frac{1}{2}(x-\mu_i)^{'}\Sigma_i^{-1}(x-\mu_i)\} \tag{3.11}$$

The parameters that characterize the GMM model are the number of Gaussians,

and for each Gaussian, its weight w_i, mean vector (μ_i) and covariance matrix (Σ_i). The mean vector has the same number of values as the number of coefficients of the feature vector and the variance matrix is a square matrix whose dimensions are the length of the feature vector. The number of components is the only parameter that has to be decided in advance and the rest are estimated during training. The model has to satisfy the constraint $\Sigma_{i=1}^{M} w_i = 1$.

The model is described by the notation in 3.12:

$$\lambda = \{w_i, \mu_i, \Sigma_i\} \qquad i = 1, ..., M. \tag{3.12}$$

The method used for estimating the parameters of a GMM is Expectation-Maximization (EM) [17]. EM algorithm is capable to calculate the estimators with a maximum likelihood criterion. It is composed by two steps that are expectation and maximization and it iterates as many times as needed until it reaches a tolerance setup to guarantee that the solution is optimal or as maximum number of iterations. The procedure of this algorithm consists in updating the values of the searched parameters with the condition that the approximation has to be more accurate after every iteration.

The estimation step calculates the expectation of the likelihood by including latent variables as if they were known and the maximization step has the objective of calculating the maximum likelihood estimators by maximizing the expected likelihood of the expectation step. The outputs from each step are the input for the other step.

The formulas used to estimate the parameters in each step are the following [16]:

Mixture Weights:

$$\bar{w_i} = \frac{1}{T} \sum_{t=1}^{T} Pr(i|x_t, \lambda) \tag{3.13}$$

Means:

$$\bar{\mu_i} = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} \tag{3.14}$$

Variances (diagonal covariance):

$$\bar{\sigma_i}^2 = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)} - \bar{\mu_i}^2, \tag{3.15}$$

where σ_i^2, x_i^2 and μ_i^2 are the variance, observation and mean vector, respectively.

The a posteriori probability is for each component i is calculated with equation 3.16:

$$Pr(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} w_k g(x_t|\mu_k, \Sigma_k)} \tag{3.16}$$

where w_i^2, Σ_i^2, x_i^2 and μ_i^2 are the weights, covariance matrix, observation and mean vector, respectively.

After the model for each class has been estimated during the training stage, classification can be applied to the incoming audio files that have to be classified. For classifying classification, the decision criterion applied is the maximum likelihood (ML). Its purpose is to determine to which of the classes the input audio segment is more likely to belong to. The equation to perform ML estimation can be written as in 3.17:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \tag{3.17}$$

where x_i^2 is the observation vector and λ represents the model.

This formula involves the products of the probability related to each frame. These terms can be multiplied because the probability is calculated in frames, and the frames are considered to be statistically independent. In order to simplify the calculation, it is desirable to perform sums instead of multiplications, which can be done working in log domain. In this case the sum the log-probabilities is calculated. The classification decision is taken based on the probabilities: the model that has the highest log-probability is output as the class to which the tested audio belongs to.

3.8 Evaluation

The evaluation of the performance is accomplished by calculating the accuracy, which measures the number of segments that have been classified correctly with respect to the total amount of test segments.

4. EXPERIMENTS AND RESULTS

4.1 Database

The database used for this project is TUT Acoustic Scenes 2016 dataset [12]. This dataset is formed by two different subsets of audio segments: a development dataset and a challenge dataset. The development dataset consist of audio files, whose labels were available for the developers for system training during the development stage in DCASE 2016. The evaluation dataset was released later without labels and was used to evaluate the systems of the developers. The scope of this procedure was that the system is developed and tested with the development data, and all the system parameters are tuned using the development data; when a high performance is achieved the system is tested with the evaluation data to determine if the system is well constructed in the sense that it has generalization properties. The evaluation data is tested only after the system is completely implemented. It comprises approximately 30% of the total amount of segments. The development dataset is divided into 4 folds in a cross-validation setput. The number of segments for each class is 78 for training and 26 for testing. All the segments have a duration of 30 seconds.

The provided audio data consists of the 15 different acoustic scenes: lakeside beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, urban park, residential area, train, and tram. For variability, recordings were done each in different locations, 3-5 minutes per location. Later, these were split into segments with alength of 30 seconds. Further information concerning to the dataset can be found in [12].

4.2 Parameters

The parameters used for this thesis have preserved the values with whom the baseline system. This means that the parameters related to the MFCC features were kept

and those related to the statistical models were varied.

MFCC were calculated with a Hamming window of 40ms and 50% overlap, 40 mel bands and the coefficients kept are the first 20, including the 0th coefficient. The window used to calculate delta and delta-delta coefficients is formed of 9 frames and the resulting feature has length of 60.

During development stage,GMMs with 8 and 16 components were estimated, considering that when clustering is performed, the number of observations belonging to each subclass is lower than those belonging to the original classes. The number of components then was decreased in order to avoid overfitting. However, the number of components used during the evaluation stage has been fixed to 16 components.

Regarding the clustering stage, two different approaches have been followed. First is a more sophisticated approach that consists in selecting the number or clusters for each class automatically. In this case, the system divides each class into the number of clusters that gives better cluster evaluation results based on the criteria of DB or CH index. Clustering using k-means was performed repeatedly with different numbers of clusters, starting from 2 to a maximum chosen number of clusters, and the number of clusters that provides best DB or CH index is selected. The number of maximum clusters has been set to 10. The number of clusters for each class is therefore selected automatically based on a clustering quality criterion. In contrast to the automatic method, the second approach is more rudimentary and consists in manual selection of the number of clusters based on the accuracy of classification obtained for each class. The number of clusters used during this study was set to 2, 3, 5 and 10 clusters. The results are compared and the evaluation is performed setting the number of clusters of each class to these that provided higher performance.

4.3 Results

This section is divided in two subsections. The first section is dedicated to the analysis of the development dataset, and the results obtained form this study is used to analyze the challenge dataset. In particular, the number of clusters that provides better performance for the development dataset are used as in input in the challenge stage to study if the performance of the system increases when optimal number of classes is selected.

4.3.1 Development dataset

The first analysis consists in a comparison of the performances for the development data when a fixed number of clusters is given to k-means algorithm. We compare one cluster, which is in fact the baseline system, with 2, 3, 5 and 10 clusters. For all the clusters the number of GMM components were set to 8 and 16 components. The results are shown in table 4.1.

Number of clusters	1 (baseline)	2		3		5		10	
GMM components	16	8	16	8	16	8	16	8	16
Beach	71.9	74.6	75.8	69.3	74.3	71.9	**77.1**	71.9	71.9
Bus	**62.0**	55.7	56.9	54.1	52.9	58.2	52.9	45.3	44.1
Cafe/restaurant	83.9	**85.4**	82.4	80.3	71.5	83.9	77.4	71.1	80.8
Car	75.7	69.1	71.7	75.5	68.8	74.1	83.3	**83.4**	78.5
City center	85.6	80.8	80.8	**86.0**	82.0	71.3	76.4	72.6	75.3
Forest path	65.9	**89.7**	73.8	72.8	75.6	76.6	73.8	66.5	72.4
Grocery store	**76.6**	63.5	63.5	67.7	67.7	59.7	61.0	55.5	55.5
Home	**79.4**	71.5	70.3	65.4	76.5	79.3	78.1	76.9	79.4
Library	**61.3**	49.9	59.4	42.2	57.7	56.6	55.4	40.4	42.0
Metro station	85.2	84.1	85.2	81.3	**86.8**	78.9	75.4	78.1	84.4
Office	**96.1**	93.4	89.7	84.3	95.0	80.0	92.1	71.7	80.2
Park	24.4	35.8	35.7	29.0	28.2	32.9	**39.3**	34.0	36.7
Residential area	75.4	75.3	75.3	76.4	80.3	79.0	80.2	**85.3**	83.8
Train	36.7	38.3	42.1	34.5	42.5	43.5	47.6	**54.9**	50.8
Tram	**89.5**	85.4	82.5	82.3	83.7	65.1	76.9	67.0	69.6
OVERALL	**71.3**	70.2	69.7	66.7	69.6	67.4	69.8	65.0	67.0

Table 4.1 *Performance for fixed number of clusters for the development data.*

The results presented in table 4.1 show the different performance reached by each number of fixed clusters. It can be observed that in some cases the highest performance is reached when no clustering is performed, and in some other cases, higher performance is reached with clustering. The best performance reached by each class is highlighted.

It can also be observed that if the number of clusters is kept invariant for all the classes, the overall performance achieved by the system is lower when clustering is made with respect to the situation when no clustering is performed. However, the performance of the system can be increased overall if a different number of clusters is selected for each class for the development data. Nevertheless, this increase must be validated with the challenge data to prove that the system has generalization properties.

The increase of the performance when the optimal number of clusters is selected for each class (values highlighted in table 4.1 is shown in table 4.2 and illustrated in figure 4.1).

Class	Baseline (%)	Innovation (%)	Improvment (%)
Beach	71.9	77.1	5.20
Bus	62.0	62.0	0.0
Cafe/Restaurant	83.9	85.4	1.5
Car	75.7	83.4	7.7
City center	85.6	86.0	2.7
Forest path	65.9	89.7	23.8
Grocery store	76.6	76.6	0.0
Home	79.4	79.4	1.3
Library	61.3	61.3	0.0
Metro station	85.2	86.8	1.6
Office	96.1	96.1	0.0
Park	24.4	39.3	14.9
Residential area	75.4	85.3	9.9
Train	36.7	54.9	18.2
Tram	89.5	89.5	0.0
OVERALL	71.3	**76.9**	5.6

Table 4.2 *Performance for the optimal number of clusters selected manually for the development data.*

A more sophisticated approach consists in selecting the number or clusters for each class automatically. In this case, the system divides each class into the number of clusters that gives better cluster evaluation results based on the criteria of DB or CH index. Table 4.3 shows the best number of clusters selected by CH index for each class, and table 4.4 shows the best number of clusters selected by the DB index.

We can notice in tables 4.3 and 4.4 that CH and DB indices give very different results: CH index shows more tendency to select 2 clusters as the best option whereas DB index selects 2 clusters in some cases and 10 in the rest. Here we can observe the importance of fixing a maximum number of clusters. DB index chose the highest possible number of clusters, because when the number is very high the observations tend to be divided each in one cluster, to minimize the within distance. It is also important to note that these indices cannot give one cluster as an output, which in some cases provides the best classification performance according to table 4.1.

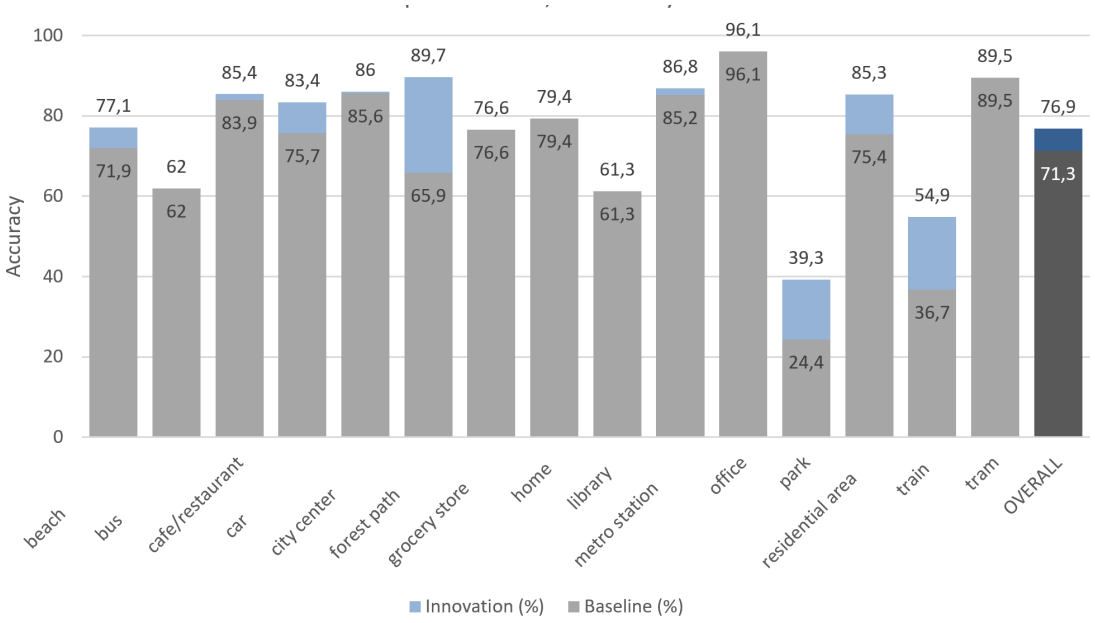

Figure 4.1 *Comparison of performance between the baseline and the innovation system for the development data. Bars in gray shows the performance of the baseline system and bars in blue shows the performance of the innovation system per class. Overall performance is shown in darker colors.*

Tables 4.5 and 4.6 shows the numerical performance of the system for the development data with CH and DB indices, respectively. It can be observed that each method works better for different classes. Some classes are improved with CH index more than with DB index and the opposite happens for other classes. Regarding the overall value for these indices, with both criteria lower performance is achieved. However, CH criterion gives better performance than DB criterion for the development dataset. Both criteria have to be analyzed for the challenge data.

4.3.2 Challenge dataset

After analysis with the development dataset has been carried out, analysis with the challenge dataset is performed. We first make a comparison of the performances for the challenge data when a fixed number of clusters is given to k-means algorithm. We compare one cluster, which is in fact the baseline system, with 2, 3, 5 and 10 clusters. For all the clusters the number of GMM components were set to 16 components. The results are shown in table 4.7. It can be observed that the results for the challenge dataset achieve higher performance that those for the development dataset. In this

Class	Fold1	Fold2	Fold3	Fold4
Beach	2	2	2	2
Bus	2	2	2	2
Cafe/Restaurant	2	2	2	2
Car	2	2	2	2
City center	2	2	2	2
Forest path	2	2	2	2
Grocery store	2	2	2	2
Home	2	2	2	2
Library	2	2	2	2
Metro station	2	2	2	2
Office	2	2	2	2
Park	2	2	2	2
Residential area	2	2	2	2
Train	2	2	2	2
Tram	2	2	2	2

Table 4.3 *Number of clusters selected by CH index for the development data.*

case, clustering with fixed number of clusters reaches higher performance than the baseline for number of clusters fixed to 2, 3 and 5. The performance decreases for 10 clusters. This fact, reinforces the idea of that a maximum number of clusters must be set when using cluster evaluation metrics. Each observation tends to be allocated in one cluster since in this case the within distance would be zero. The best performance is achieved for 2 clusters.

The second approach consist in selecting a different number of clusters for each class. The results obtained for the development data have to be validated with the challenge data to verify if the system has generalizing properties. The performance for the optimal cluster obtained with the challenge dataset can be observed in table 4.8 and illustrated in figure 4.2.

A more sophisticated approach consists in selecting the number or clusters for each class automatically. In this case, the system divides each class into the number of clusters that gives better cluster evaluation results based on the criteria of DB or CH index. Tables 4.9 and 4.10 shows the numerical performance of the system for the CH and DB criteria, respectively. It can be observed that each method works better for different classes. Some classes are improved with CH index more that with DB index and the opposite happens for other classes. Regarding the overall value for these indices, with both criteria higher performance is achieved with respect to

Class	Fold1	Fold2	Fold3	Fold4
Beach	2	2	2	2
Bus	10	10	10	10
Cafe/restaurant	2	10	10	10
Car	2	2	10	2
City Center	2	2	2	2
Forest path	2	2	2	2
Grocery store	10	10	10	10
Home	2	2	2	2
Library	2	2	2	2
Metro station	2	2	2	2
Office	2	2	2	2
Park	2	2	2	2
Residential area	2	2	2	2
Train	2	2	2	2
Tram	2	2	10	2

Table 4.4 *Number of clusters selected by DB index for the development data.*

the baseline system. Both criteria have overall equal performance.

Based on the previous results, a comparison can be done between all the methods. table 4.11 shows the level of accuracy for each method and figure 4.3 shows the results graphically.

Class	CH
Beach	75.9
Bus	55.7
Cafe/restaurant	82.4
Car	71.7
city center	80.8
Forest path	73.8
Grocery store	66.2
Home	68.9
Library	56.9
Metro station	85.2
Office	89.7
Park	36.9
Residential area	72.7
Train	42.1
Tram	83.9
OVERALL	69.5

Table 4.5 *Performance for the number of clusters selected by CH index for the development data.*

Class	DB
Beach	77.1
Bus	35.0
Cafe/restaurant	65.5
Car	87.2
city center	80.8
Forest path	73.8
Grocery store	48.0
Home	71.4
Library	44.9
Metro station	85.2
Office	76.5
Park	34.3
Residential area	71.7
Train	52.1
Tram	77.0
OVERALL	65.4

Table 4.6 *Performance for the number of clusters selected by DB index for the development data.*

Number of clusters	1 (baseline)	2	3	5	10
Beach	84.6	76.9	80.8	76.9	76.9
Bus	88.5	100.0	100.0	92.3	96.2
Cafe/Restaurant	69.2	61.5	61.5	61.5	61.5
Car	96.2	92.3	84.6	80.8	76.9
City center	80.8	88.5	92.3	92.3	80.8
Forest path	65.4	88.5	84.6	88.5	84.6
Grocery store	88.5	84.6	73.1	69.2	69.2
Home	92.3	88.5	84.6	100.0	96.2
Library	26.9	42.3	23.1	23.1	19.2
Metro station	100.0	88.5	88.5	92.3	92.3
Office	96.2	92.3	100.0	100.0	96.2
Park	53.8	69.2	69.2	73.1	73.1
Residential area	88.5	80.8	80.8	80.8	73.1
Train	30.8	46.2	50.0	69.2	73.1
Tram	96.2	88.5	92.3	80.8	80.8
OVERALL	77.2	**79.2**	**77.7**	**78.7**	76.7

Table 4.7 *Performance for fixed number of clusters for the challenge data.*

Class	Baseline (%)	Innovation (%)	Improvment (%)
Beach	84.6	76.9	-7.7
Bus	88.5	88.5	0.0
Cafe/Restaurant	69.2	61.5	-7.7
Car	96.2	76.9	-19.2
City center	80.8	92.3	11.5
Forest path	65.4	88.5	23.1
Grocery store	88.5	88.5	0.0
Home	92.3	92.3	0.0
Library	26.9	26.9	0.0
Metro station	100.0	88.5	-11.5
Office	96.2	96.2	0.0
Park	53.8	73,1	19.2
Residential area	88.5	73.1	-15.4
Train	30.8	73.1	42.3
Tram	96.2	96.2	0.0
OVERALL	77.2	**79.5**	2.3

Table 4.8 *Performance for the optimal number of clusters for the challenge data.*

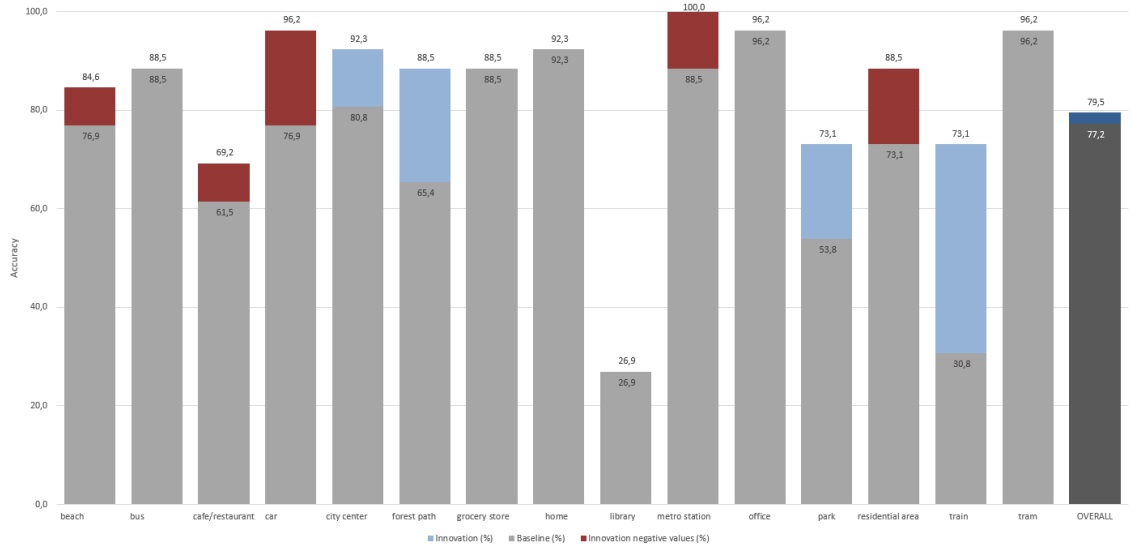

Figure 4.2 *Comparison of performance between the baseline and the innovation system for the challenge data. Bars in gray shows the performance of the baseline system and bars in blue shows the performance of the innovation system per class. Overall performance is shown in darker colors.*

Class	CH
Beach	76.9
Bus	100.0
Cafe/Restaurant	61.5
Car	92.3
City center	88.5
Forest path	88.5
Grocery store	84.6
Home:	88.5
Library	42.3
Metro station	88.5
Office	92.3
Park	69.2
Residential area	80.8
Train	46.2
Tram	88.5
OVERALL	**79.2**

Table 4.9 *Performance for the number of clusters selected by CH index for the challenge data.*

Class	DB
Beach	76.9
Bus	96.2
Cafe/Restaurant	61.5
Car	80.8
City center	88.5
Forest path	92.3
Grocery store	73.1
Home	96.2
Library	15.4
Metro station	92.3
Office	100.0
Park	76.9
Residential area	80.8
Train	76.9
Tram	80.8
OVERALL	**79.2**

Table 4.10 *Performance for the number of clusters selected by DB index for the challenge data.*

Method	Accuracy
Baseline	77.2
2 clusters	79.2
3 clusters	77.7
5 clusters	78.7
10 clusters	76.7
CH	79.2
DB	79.2
Optimal number of clusters	**79.5**

Table 4.11 *Comparison of performance between all the methods for the challenge data.*

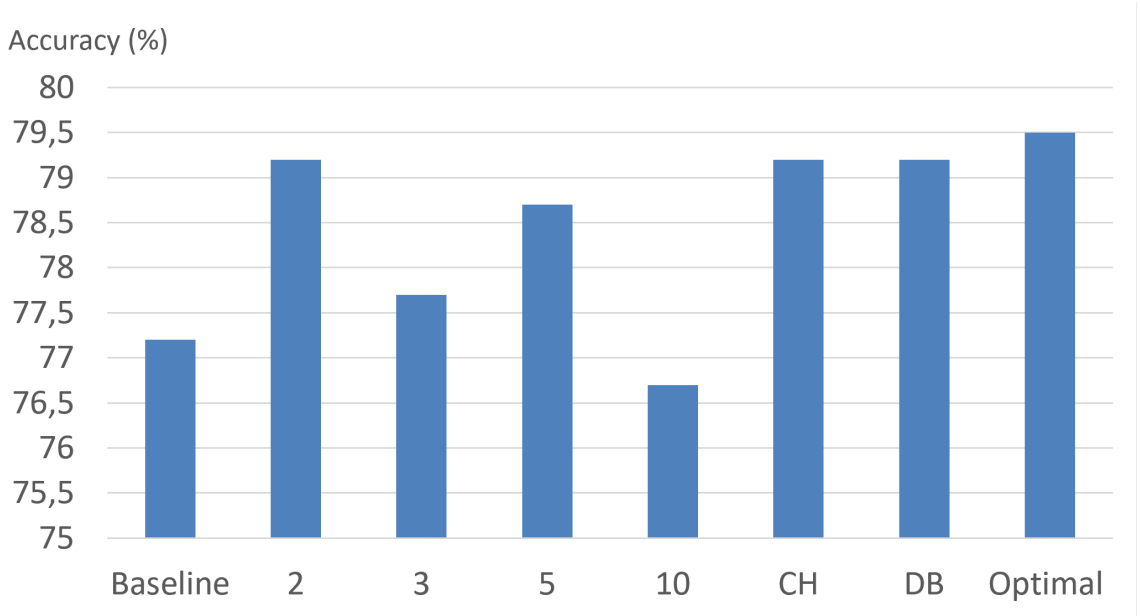

Figure 4.3 *Comparison of performance between all the methods for the challenge data.*

5. CONCLUSIONS

This thesis consists in the extension of a baseline system capable to recognize the acoustic environment in which an audio is recorded. The different environmental acoustic scenes are predefined in advance. The extension is based on the motivation of studying if unsupervised learning brings improvements in terms of higher accuracy to a system that is based in supervised learning. Therefore, this projects aims to combine both supervised and unsupervised learning for solving a problem of ASC.

During the chapter dedicated to the state of the art, a bibliographical revision has been done. This bibliographic review includes the methods that have been used previously for ASC. Most of the approaches follows the same structure that consists in performing a training stage where a model is constructed based on the observation, followed by a testing stage that performs the classification of input audio signals to output the label their respective classes. Most of the systems included in the review, build the model based on acoustic features that have been previously extracted. The methods implemented by the baseline pipeline have been preserved in this project, which include extraction of MFCC features and statistical models based on GMM. The method selected for the innovation part has been k-means clustering.

The empirical part of this project is, therefore, focused in the innovation part. The system has been built with the purpose to serve as a tool to determine the influence of the clustering and its tuning possibilities. The experiments have been performed firstly with the development dataset and the results obtained have been validated with the challenge dataset aiming to verify that the system is capable to generalize its results. It has been proved that clustering brings higher performance to the system by analyzing three different approaches.

First approach is based on the idea of selecting the same number of clusters for all classes. Experiments have been done with 2, 3, 5 and 10 clusters. The values 2, 3 and 5 have improved the performance. Value 10 has proved to reduce the performance. It can be deduced from these results that large number of clusters

reduces the performance of the system. The reason is that when the number of clusters is large, each cluster tends to contain a single value. The highest level of accuracy obtained has been 2% higher with respect to the baseline and corresponds to 2 clusters.

Second approach consist in the idea of selecting manually the number of clusters that proved to give better performance for each class during the development stage and choosing the number of clusters for each class during the challenge stage according to those results. The highest performance obtained by the system has been achieved by using this method. The performance has increased 2.3% with respect to the baseline.

Third approach is more sophisticated than the previous ones and includes cluster evaluation based on clustering metrics. This method permits grouping each class in different number of clusters automatically. The indices tested have been BD and CH and both have proved to provide the same level of accuracy for the challenge data. The level of accuracy has been higher than when fixing the number of clusters but lower than when selecting manually the number of clusters. The performance has increased 2% with respect to the baseline.

BIBLIOGRAPHY

[1] N. Ibáñez Sáez *et al.*, "Técnicas de procesado de imagen para el seguimiento de objetos desde vehículos aéreos no tripulados," 2015.

[2] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[3] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[4] B. Defréville, F. Pachet, C. Rosin, and P. Roy, "Automatic recognition of urban sound sources," in *Audio Engineering Society Convention 120*. Audio Engineering Society, 2006.

[5] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*. IEEE, 2005, pp. 158–161.

[6] P. Guyot, J. Pinquier, and R. André-Obrecht, "Water sound recognition based on physical models," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 793–797.

[7] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE, 1994, pp. 85–90.

[8] Y. Xu, W. J. Li, and K. K. Lee, *Intelligent wearable interfaces*. John Wiley & Sons, 2008.

[9] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. J. Mataric, "Where am i? scene recognition for mobile robots using audio features," in *Multimedia and Expo, 2006 IEEE International Conference on*. IEEE, 2006, pp. 885–888.

[10] C. Landone, J. Harrop, and J. Reiss, "Enabling access to sound archives through integration, enrichment and retrieval: The easaier project." in *ISMIR*, 2007, pp. 159–160.

[11] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on.* IEEE, 2013, pp. 1–4.

[12] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Signal Processing Conference (EUSIPCO), 2016 24th European.* IEEE, 2016, pp. 1128–1132.

[13] D. Pascual, F. Pla, and S. Sánchez, "Algoritmos de agrupamiento," *Método Informáticos Avanzados*, 2007.

[14] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[15] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[16] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, pp. 827–832, 2015.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.