Ricardo Miguel Pontes Leonardo

Bachelor of Science in Biomedical Engineering

Contextual Information based on Pervasive Sound Analysis

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Biomedical Engineering

Adviser: Hugo Filipe Silveira Gamboa, Professor Auxiliar, Faculdade de Ciências e Tecnologias, Universidade Nova de Lisboa

Examination Committee

Chairperson:	Prof. Dr. Carla Maria Quintão Pereira
Raporteur:	Prof. Dr. Ricardo Nuno Pereira Verga e Afonso Vigário
Member:	Prof. Dr. Hugo Filipe Silveira Gamboa

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2017

Contextual Information based on Pervasive Sound Analysis

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

"Se perderes a direção da lua,
Olha a sombra que tens colada aos pés"

Rio Grande, "Senta-te aí"

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Hugo Gamboa, for welcoming me at *Fraunhofer AICOS* and being an excellent advisor throughout the whole project. Thank you for the great guidance and the dedication, as well as for making me grow as a professional. Thank you also for reinforcing my love for data science, both before as a professor and now as a supervisor.

I would also like to thank *Associação Fraunhofer Portugal Research* for all the support and the opportunity of working on my thesis in a company environment. A word of appreciation in particular to everyone in the Lisbon, team for being the living proof that a work environment can and must be fun in order to be productive. A very special thanks to Marilia Barandas for all the patience for me and the constant support, your help was invaluable. Also a word of gratitude for my colleague David Melo for his great friendship and for accompanying me throughout this experience.

Thanks to all my college friends, I could fill a book with my gratitude for all of you. Thank you for all the laughs, and for all those silly moments that make life worth living. I couldn't have done this without you.

And last but not least, I thank my family for always being there for me, both in moments of joy and hardship. The most heartfelt thanks to my parents, to whom I owe everything I am and who always believed in me. Words are not enough to describe how thankful I am. I love you all.

Abstract

In recent times, there has been a continuous increase in the ubiquity, processing power and sensing capabilities of modern smartphones. This has made possible the emergence of new technologies that allows users to keep track of information regarding their health, activities and location, even in indoor places were GPS signal is not available. These technologies typically rely on fusing and processing information coming from multiple sensors, such as the accelerometer or the magnetometer.

This thesis proposes a framework for indoor location and activity recognition from new source of information: the sound perceived through the device's microphone. It does so by extracting information relative to the user's position and activities through machine learning and audio processing techniques.

In the context of indoor location, the proposed SoundSignature algorithm allows the device to learn from labeled data and predict the location it is in. These locations may be different rooms or distinct regions of large places, such as open spaces.

Another proposed algorithm, SoundSimilarity, further refines this positioning by comparing the sound signals from two or more devices in real time. A novel audio similarity metric identifies if the devices are close to one another, mitigating the potential errors of the SoundSignature algorithm. This also has many other use cases, such as detecting proximity between the user and devices.

Finally, the Activity Monitoring algorithm allows the device to learn from labeled data to recognize the activity the user is performing. This information may be also used to further refine the location algorithm by recognizing location-dependent activities such as the closing of doors or watching television.

Keywords: Indoor Location, Human Activity Recognition, Machine Learning, Signal Processing, Audio Analysis

Resumo

Nos últimos tempos, tem havido um aumento contínuo da ubiquidade, poder de processamento e capacidade sensorial nos *smartphones* modernos. Isto possibilitou o aparecimento de novas tecnologias que permitem aos utilizadores monitorizar informações sobre a sua saúde, atividades e localização, mesmo em zonas interiores onde o sinal GPS não está disponível. Estas tecnologias geralmente dependem de processamento de informação proveniente de vários sensores tais como o acelerómetro ou o magnetómetro.

A presente tese propõe uma *framework* para localização em zonas interiores e reconhecimento de atividade através de uma nova fonte de informação: o som percebido através do microfone do dispositivo. Para tal, o sinal é processado com técnicas de *machine learning* e de processamento de sinal.

No contexto da localização em espaços interiores, o algoritmo *SoundSignature* permite que o dispositivo aprenda com dados rotulados e identifique a localização em que se encontra. Esses locais podem ser divisões diferentes ou regiões distintas de locais amplos.

Outro algoritmo, *SoundSimilarity*, compara os sinais de som de dois ou mais dispositivos em tempo real com uma métrica de similaridade de áudio para identificar se estes estão próximos uns dos outros. Isto não só ajuda a mitigar potenciais erros do algoritmo anterior como também pode ser aplicado noutros casos, tal como detetar a proximidade entre o usuário e outros dispositivos.

Finalmente, o algoritmo *Activity Monitoring* permite que o dispositivo aprenda com dados rotulados para identificar a atividade que o utilizador está a realizar. Esta informação pode também ser utilizada para para localizar o utilizador ao reconhecer atividades dependente de Esta informação pode também ser usada para auxiliar os algoritmos de localização, reconhecendo atividades dependentes da localização tais como fechar portasou ver televisão.

Palavras-chave: Localização *indoor*, Reconhecimento de Atividades Humanas, Aprendizagem Automática, Processamento de Sinal, Análise de Áudio

Contents

List of Figures

List of Tables

Acronyms

AUC	Area Under the Curve.
DTW	Dynamic Time Wrapping.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
HAR	Human Activity Recognition.
MAS	Measurement for Audio Similarity.
MFCC	Mel Frequency Cepstral Coefficients.
OvO	One-vs-one.
OvR	One-vs-rest.
rbf	radial basis function.
ROC	Receiver Operating Characteristic.
SVM	Support Vector Machines.

Chapter 1

Introduction

1.1 Context and Motivation

In recent times, there have been major developments in the smartphone industry. Their ever increasing processing power and continuous connection to the Internet have made them essential part of our lives, both as work and entertainment tools. This, allied to their also ever increasing sensing capabilities, increased the market's interest in Human Activity Recognition (HAR) technologies.

HAR is a field of computer science that integrates sensor data with machine learning algorithms to recognize a wide range of human activities such as brushing teeth or walking [4]. This creates new technologies that allow users to better keep track their daily habits and improve their lifestyle. Other use cases include assisting caretakers in the monitoring of elderly people or reminding rehabilitation patients to execute their prescribed exercises.

These technologies typically rely on tracking the device's movements through sensors such as the accelerometer or the magnetometer. However, the use of these sensors require that the device is attached to the user, commonly in specific body locations. Furthermore, relying on movement exclude the detection of certain activities that do not incur in it, such as talking or watching television.

Other field where smartphones are used everyday is in navigation systems and location-dependent services. In this context, the most well known and widespread technology is Global Positioning System (GPS), included in the Global Navigation Satellite System (GNSS). However, while this system's precision and satellite coverage are typically sufficient for outdoor applications, this is not the case when the user is inside a building. The presence of walls and ceilings between the user and the satellites greatly attenuates the latter's signal, and the reduced scale of typical paths in buildings compared to outdoor

routes create a demand for better precisions.

These deficits in both HAR and indoor location created a need for new sources of information for these activities. As such, in the present work we study the analysis of pervasive sound as a source of contextual information in these two contexts.

1.2 Objectives

As the responsible for one of the five traditional senses, our auditory system is constantly picking up large amounts of information about our surroundings. This information may be event-dependent, such as walking or the closing of doors, or location-dependent, such as the humming of computers or air conditioning systems. The aim of this thesis is to translate this innate ability of ours to any microphone equipped mobile device.

This thesis presents an original framework developed for extracting information from sound regarding the user's position and the activities they are performing. For this purpose, three algorithms were created:

1. **SoundSignature:** Recognizes the location the user is in. These locations consist of different rooms or distinct regions of large spaces. Relies on machine learning techniques to learn from labeled data.

2. **SoundSimilarity:** Extracts a novel measurement of audio similarity from the sounds perceived in real time by two or more devices, allowing it to identify if these are in the same location.

3. **Activity Monitoring:** Analyses pervasive sound to recognize activities the user performs. Similarly to the SoundSignature algorithm, relies on machine learning techniques to learn from labeled data.

1.3 Literature Review

Multiple alternative solutions have been proposed in the literature for indoor positioning systems. Many leverage signals transmitted between beacons and the device to be located, namely Radio Frequency signals, either Wi-Fi [15] or Bluetooth [6], infrared signals [10], ultrasound [2], visible light [21], among others. By estimating the distance of the device to each beacon through metrics such as received signal strength (RSS) [31] and time difference of arrival (TDoA) [18], trilateration may be used to locate the device. Other methods include using these metrics for fingerprinting techniques [3]. However, most of current systems rely on infrastructure, leading to elevated setup and maintenance costs.

Some infrastructure-free solutions have also been proposed, such as using pervasive signals such as ambient light [16] and perturbations in the Earth's magnetic field for fingerprinting techniques [17]. Novel systems integrate many of the afore mentioned developments with inertial tracking and map data to locate the user indoors [9]. An

existing sound-based solution by Stephen P. Tarzia et al. [27] uses the power spectra the audio signal as an acoustic fingerprint to differentiate between different rooms [27].

Furthermore, Jun-Wei Qiu and Yu-Chee Tseng [22] have shown that meetings between two or more users may be used to calibrate their respective potential locations. Sound-based proximity detection was achieved through use of the normalized cross-correlation coefficient between the spectra of the compared signals [23].

Regarding the context of HAR, multiple solution have been developed about identifying the user's activities from recorded sound. The most transversal element to them is the use of MFCC, a group of features based on the human perception of pitch. These features are commonly used in speaker [28] and speech recognition [12].

Yi Zhan et al. [34] employ these features for HAR. The authors split a sound segment into windows of smaller length and for each of these MFCC were extracted. The resulting matrix is then compared to previously recorded and labeled templates through the use of a Dynamic Time Wrapping (DTW) algorithm, achieving an accuracy of 92.5%.

Johannes A. Stork et al. [26] also employ MFCC as features, but instead use a Random Forests algorithm for classification. This method achieved an accuracy of 85.8%.

Yao Yang et al. [32] use features such as Spectral Centroid and Spectral Roll-off to preliminarily determine if the recorded sound consists of speech, music or a human activity such as brushing teeth. After this step MFCC are once again employed to determine the activity.

Yi Zhan and Tadahiro Kuroda [35] released another study regarding HAR, this time employing features extracted from a continuous wavelet transform with Haar wavelets. To these they called "Haar-like features". The chosen classifier was Hidden Markov Models and the achieved accuracy was 96.9%. For comparison, the authors used the same algorithm and dataset but recurring to MFCC instead of these "Haar-like features"and the accuracy dropped to 88.7%.

Jia-Ching Wan et al [30] elaborate on previous work by applying Individual Component Analysis to the MFCC, generating artificial features with high statistical independence between them. For classification, a Support Vector Machines (SVM) classifier was used, achieving an 86.7% accuracy.

Etto L. Salomon et al. made a thorough comparison of features and classifier algorithms in the context of sound-based HAR. Notably, the authors followed the protocol described in [34] and got poor results. As such, the "Haar-like features"were preliminarily discarded. The study concluded that the best results were obtained with MFCC as features and SVM as classifier.

Daniel Kelly and Brian Caulfield [13] used an algorithm based on MFCC and SVM for distinguishing the sound of cutlery from the sound of tap water, achieving an accuracy of 96.9%. Applying the same algorithm to discriminating between speech, music and environmental sounds yelded an accuracy of 88.7%.

Finally, C. E. Galván-Tejada employed MFCC along with statistical features such as standard deviation and median to distinguish between quotidian sounds. Two classifier

algorithms were compared: Random Forests, which yielded an accuracy of 81.4%, and Neural Networks, with an accuracy of 77.7%.

In conclusion, current infrastructure-free indoor location algorithms rely on fingerprinting techniques or inertial sensors, the latter being irrelevant in the current work. Furthermore, the features most prevalent in current literature regarding sound classification are MFCC and features extracted from the frequency spectrum. Likewise the most commonly used classifier algorithm is SVM.

1.4 Thesis Overview

The present chapter introduced the motivation behind the development of the current work, as well as the main objectives and a review of the literature regarding the subject at hand. Chapter 2 provides some theoretical concepts about audio processing and machine learning, both relevant to this work.

Chapter 3 presents the developed framework, describing the methodologies used and explaining the deliberations made in their design.

Chapter 4 show the results obtained, along with the composition and acquisition methods of the used datasets. Chapter 5 present the main conclusions of this thesis and guidelines for future work.

Finally, appendix A shows the routes used for recording the SoundSignature dataset, which will be relevant in section 4.1.2.

Chapter 2

Theoretical Background

2.1 Machine Learning

In computer science, machine learning can be defined as the branch that gives computers the ability to learn without being explicitly programmed [33]. It does so by extracting features from a training set of data, allowing the algorithm to learn the underlying model and thus classify future inputs.

According to the data available to be used, there are various categories of machine learning algorithms:

- **Supervised learning**: The algorithm is presented with a set inputs and their respective desired outputs, or labels. The goal is to generate a model that maps the inputs to the outputs.
- **Unsupervised learning**: The algorithm is presented with unlabeled inputs. The goal is to separate the inputs into clusters based on their similarity.
- **Semi-supervised learning**: The algorithm uses both a small amount of labeled data and a large amount of unlabeled data.
- **Reinforced learning**: Instead of learning from discrete samples of data, the algorithm interacts with a dynamic environment in which it must perform a certain goal, such as winning at a game of checkers or driving a car.

The problems at hand will rely on labeled data, and as such we will be focusing on supervised learning.

2.1.1 Preprocessing

In order to apply machine learning algorithms to continuous streams of data, we must first segment it into individual inputs. This is typically done by dividing the signal into windows of equal length, which may or may not have some overlap between them.

The length of the window will have an impact on the performance of the algorithm. On the one hand, smaller windows will increase the temporal resolution and reduce computational costs. On the other hand, larger window lengths will provide larger amounts of data per sample, allowing better recognition.

Furthermore, increasing the size of the overlap will provide the algorithm a larger training set, at the cost of some redundancy between samples.

2.1.2 Feature Extraction

Features can be defined as properties extracted from certain input. These will be used as the input for a classifier algorithm, whether it is for training, testing or usage. They can be either continuous (such as temperature) or discrete (such as rainy or sunny), and can be classified according to the domain they operate in, such as time, statistical or frequency.

Different features may have different means standard deviations, which may lead to giving more importance to some features than others when training a classifier algorithm. Because of this, it is important to standardize all features so that they have mean $\mu = 0$ and standard deviation $\sigma = 1$. This is achieved through the following expression:

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{2.1}$$

Where x is a feature, μ its mean, σ its standard deviation and x_{norm} the resulting standardized feature.

2.1.3 Classification

A classifier is an algorithm that maps input data to a category. It does so by generating a model from a training set containing observations whose category is known. It is on the basis of these that supervised learning are built upon.

There are multiple classifier algorithms. The choice of the algorithm will depend on various parameters such as the kind of data, number of observations and number of individual categories.

Some examples of common classification algorithms are:

- The **K-Nearest Neighbors** algorithm finds the closest K nearest instances in the training set and classifies it according to the most frequent class among these. K is an integer, usually small in relation to the size of the training set. Larger values for K make the algorithm more robust to noise [5], but make the decision boundary less defined. This algorithm is computationally fast in the training phase but slow in classifying inputs.

- The **Naive Bayes** classifier assumes all features are independent of each other, and under this assumption calculates the probability of an input being each class given its features. This is done through the Bayes' theorem:

$$P(C_k|x) = \frac{P(C_k)P(x|C_k)}{P(x)} \tag{2.2}$$

  Where C_k is the k-th class and x the feature vector.

- **Decision Trees** create a model based nodes connected by branches in a tree-like fashion. Each node contains a simple rule pertaining to a single feature. An input starts at the top node and travels down the branches guided by these rules until a terminal node is reached. These terminal nodes indicate the class the input is classified with. Decision trees are generated by computing at each node which splits yield the most information. This can be done through a number of criteria.

- **Support Vector Machines** find the hyperplanes in the feature space that best separate the training set into their labels..

- **Ensemble Methods** create multiple independent classifiers. Their predictions are then used to achieve a final result. An example of such a classifier is **Random Forests**, where multiple decision trees are created from different subsets of the dataset. Each classifier's prediction counts as a vote towards the predicted class. The final result is the class with the most votes.

- **Neural Networks** are composed of elements called perceptrons that mimic the functioning of neurons.

2.1.3.1 Support Vector Machines

Being the most prevalent classifier algorithm in the reviewed literature, in the present work we chose to use SVM.

SVM are binary classifiers, meaning that they can only discern between two classes. If we consider an n-dimensional space where each dimension relates to a feature, this classifier finds the hyperplane $\vec{w} \cdot \vec{x} + b = 0$ that best separates the two classes.

For a given set of linearly separable points $\vec{x_i}$ in the feature space that map to two classes $y_i = 1$ or $y_i = -1$, we can assume that the margin between them can be delineated by two parallel hyperplanes $\vec{w} \cdot \vec{x} + b = 1$ and $\vec{w} \cdot \vec{x} + b = -1$, as shown in figure 2.1. We want these two hyperplanes to follow two conditions:

- The distance between them, given by $\frac{b}{\|w\|}$, must be maximum. This means we want to minimize $\|w\|$

- The data points cannot fall into the margin between them. We can impose that by adding the restrictions $\vec{w} \cdot \vec{x_i} + b \geq 1$ and $\vec{w} \cdot \vec{x_i} + b \leq -1$. This can be simplified into $y_i(\vec{w} \cdot \vec{x_i} + b) \geq 0$.

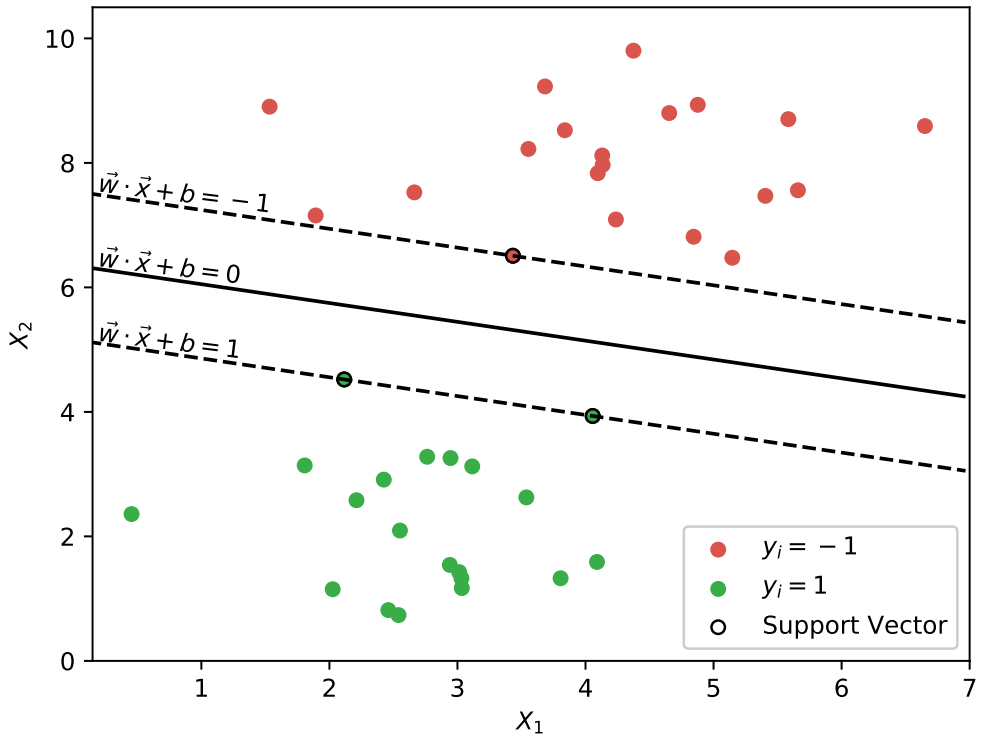


Figure 2.1: Bidimentional illustration of Support Vector Machines. $\vec{w}\cdot\vec{x}+b=0$ indicates the found hyperplane that ideally separates the two classes; $\vec{w}\cdot\vec{x}+b=1$ and $\vec{w}\cdot\vec{x}+b=-1$ delineate the margin between the classes. The points of different colors indicate samples of different classes, and the outlined ones represent the used support vectors.

While minimizing $\|\vec{w}\|$ under this restriction finds the ideal solution for fully linearly separable datasets, this is not always the case. To extend SVM to non linear separable data, instead of applying the last restriction we introduce a hinge loss function based it. This will also be an expression we want to minimize:

$$\max(0, 1 - y_i(\vec{w}\cdot\vec{x}_i + b)) \tag{2.3}$$

This value of this expression will be equal to zero if the point is on the right side of the plane and outside the margin. Otherwise, it will be proportional to the distance between the point and the margin.

Given these two expressions we want to minimize, the resulting hyperplane parameters can be obtained by minimizing the following expression:

$$\frac{1}{n}\left[\sum_{i=0}^{n-1}\max(0, 1 - y_i(\vec{w}\cdot\vec{x}_i + b))\right] + \lambda\|w\|^2 \tag{2.4}$$

Where λ represents the trade-off between increasing the size of the margin and ensuring that the $\vec{x}_i$ lie on the correct side of the margin. This expression is only affected by

points either inside or bordering the margin. To these we call support vectors.

By minimizing this expression we obtain values for $\vec{w}$ and b, obtaining the hyperplane $\vec{w} + b = 0$ that functions as a decision boundary. Inputs will be classified according to in what side of this hyperplane they are on.

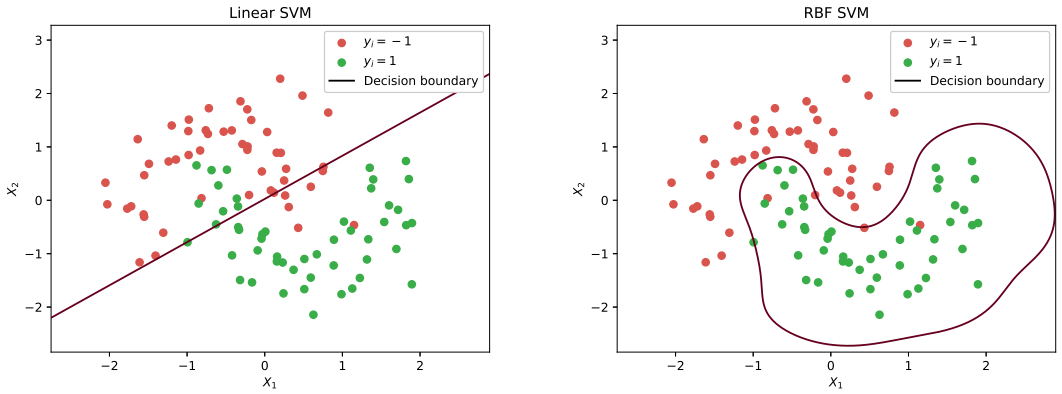
Additionally, a probability estimation of an input being classified can be obtained by how far it is from the decision boundary.

2.1.3.2 Kernel trick

SVM does its computations under the pretension that the classes are linearly separable, and as such the decision boundaries are linear. Non linear separations can be achieved by translating the feature space to higher dimensions, minimizing this expression and translating back to the original number of dimensions. This can be easily simplified by replacing every inner product $\vec{x} \cdot \vec{y}$ with a kernel function $k(\vec{x}, \vec{y})$. An example of such function is the radial basis function (rbf) kernel:

$$k(f, g) = exp(-\gamma(\vec{x} - \vec{y})) \tag{2.5}$$

Where γ is a adjustable parameter greater than 0. The smaller this parameter is, the smoother will be the decision boundary. Figure 2.2 shows the difference between decision boundaries found with and without the kernel trick.



(a) Decision boundary found with linear SVM. (b) Decision boundary found with SVM using the rbf kernel.

Figure 2.2: Comparison of the decision boundaries found in a bidimensional feature space with and without using the kernel trick.

2.1.3.3 Extension to multiclass problems

To extend binary classification algorithms to multiclass problems with k classes there are two common procedures:

- **One-vs-rest (OvR)** creates k classifiers, or one per class. Each classifier considers the samples from that class as positive and the remaining as positive. The chosen class will be the one whose classifier reports the largest confidence score.
- **One-vs-one (OvO)** creates $k(k-1)/2$ classifiers, one for each possible pair of classes. The prediction of each classifier will vote towards the overall prediction.

2.1.4 Feature Selection

While many different features may be extracted from the data, many may be irrelevant to the problem at hand. Some may have a high degree of redundancy between themselves, and as such we may keep only one of them without any significant loss of information. Others may be uncorrelated to the desired output, and therefore may also be removed.

A larger number of features leads not only to worse computational performance but also classifier accuracy, due to the curse of dimensionality. The curse of dimensionality denotes that as the number of dimensions increase, the amount of possible data combinations increases exponentially, and as such an exponentially larger dataset is needed to cover all the possibilities. Therefore, if the dataset is not large enough to support the amount of features extracted from it, the classifier will have poor performance as it might see many combinations of data it has never seen before.

Furthermore, a lower number of features allow the classifier to better generalize the model, and thus reduce overfitting.

Due to these motives, several feature selection algorithms have been designed. These allow automatically select the most relevant features from a dataset and discard the rest, increasing both computational and classifier performance. These algorithms may be classified into three types:

- **Wrapper methods** create multiple subsets of features. These are used to train the classifier algorithm selected according to an objective function, such as the resulting accuracy score. These are usually computationally intensive, but yield the best feature subset for the particular classifier in use.
- **Filter methods** compute measures of correlation between features or between features and the desired output. While these methods are typically less computationally intensive, the resulting feature subset is not tuned to the specific classifier algorithm in use, and thus result in lower predictive ability.
- **Embedded methods** make use of the classifier's own training routine for feature selection.

Because in the present work computational performance in training time was not an issue, we employed Forward Feature Selection [11], a wrapper method. This method recursively adds the best features to a candidate set until they no longer improve accuracy.

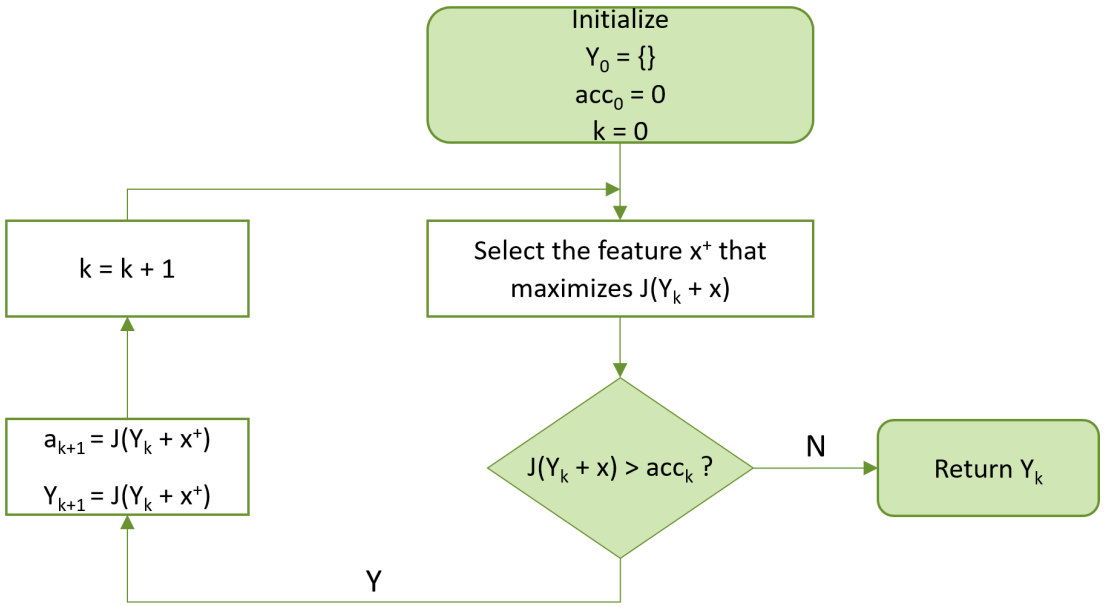


Figure 2.3: Fluxogram explaining the Sequential Forward Feature Selection algorithm

This algorithm is represented in the fluxogram in figure 2.3 and is described by the following steps:

1. Start with an empty feature set $Y_0 = \{\}$, an accuracy $a_0 = 0$, an objective function J and $k = 0$;
2. Select the feature x^+ that maximizes $J(Y_k + x)$;
3. If $J(Y_k \cup \{x^+\}) > a_k$, update $Y_{k+1} = Y_k \cup \{x^+\}$, $a_{k+1} = J(Y_k \cup \{x^+\})$ and $k = k + 1$ and go back to *2.*, otherwise continue;
4. Keep only the feature set Y_k and discard the rest.

The objective function J is a function that returns a value that quantifies the performance of the algorithm. In the present work we chose to use accuracy computed with 10-fold stratified cross-validation.

2.1.5 Validation

After a classifier has been trained, its performance must be evaluated. This is usually done by testing the resulting classifier on a dataset and observing if the resulting predictions match the original labels.

We cannot, however, use the same data we used in training for testing. Even though if the algorithm correctly classifies the data it was trained on, the classifier could show poor performance when shown new data. This is because the model might be overly

adapted to the examples presented, and thus unable to properly generalize. To this we call overfitting.

To prevent the effects of overfitting from showing in the validation phase, we must come up with more clever ways to make use of our dataset. To this end we recur to cross-validation, where part of the dataset is used to train the classifier and the other part is used to test it. There are multiple methods of performing this:

- **Holdout method:** An arbitrary percentage of samples is randomly assigned to the training set, while the remaining is used for testing. The training set is usually larger than the testing set.
- **K-fold cross-validation:** The dataset is randomly split into K equal sized subsets or folds. For i < K, the i-th fold is used for testing and the remainder data for training.
- **Stratified K-fold cross-validation:** Similar to K-fold cross-validation, but each layer has approximately the same ratio between classes as the full dataset. This is done to prevent class imbalance in each fold.
- **Leave-One-Out cross-validation:** Similar to K-fold cross-validation, but with K equal to the number of folds. This results in folds with single samples each.
- **Leave-One-Group-Out cross-validation:** Once again similar to K-fold cross-validation, but the folds are separated according to metadata such as the day of recording, user, position, *etc*.

To visualize the classifier's performance it is common to use a **Confusion Matrix**. This is an $N \times N$ matrix where the number in position (i, j) corresponds to the number of samples labeled as class i predicted as class j. Therefore, the number of correct predictions will be equal to the sum of the values positions where $i = j$. The accuracy rating can be obtained according to the following expression:

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of samples}} \tag{2.6}$$

Another way to represent data is to divide each row in the confusion matrix by its sum. In this case the number in position (i, j) corresponds to the proportion of samples labeled as class i predicted as class j. This is called a **Normalized Confusion Matrix**.

2.2 Sound Analysis

Sound is a vibration that consists on the propagation of pressure waves along a transmission medium that can be either solid, liquid or gaseous. When propagated over a liquid or gaseous medium, such as air, the vibration is longitudinal to the propagation. This creates regions of compression, where the pressure is higher, and rarefaction, where the pressure is lower.

An audio signal can be obtained through means of a microphone, a transducer that converts these sound waves into an electrical signal proportional to the pressure. Figure 2.4 illustrates a comparison between a sound wave and its representation as an audio signal.

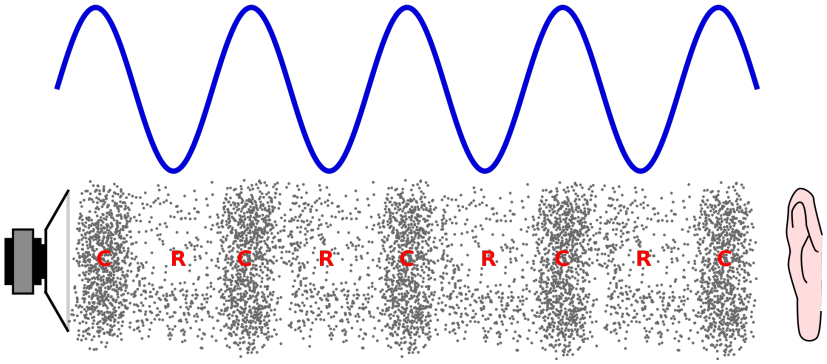


Figure 2.4: Comparison between a sound wave and its representation as an audio signal. Below is the propagation of a sound wave where the dots represent air particles, C zones of compression and R zones of rarefaction. Above is the representation of this sound as an audio signal. (How to cite image from wikimedia commons?)

The electrical signal obtained from a microphone is continuous. However, a computer can only store and process discrete data, meaning we have to quantize the signal as in figure 2.5. Quantization is the process of mapping a signal from continuous to discrete values both in time and amplitude, returning a discrete signal. Quantization in time is characterized by the sampling frequency, which indicates the number of values per second, and the quantization in amplitude is characterized in bit-depth, which defines the number of bits required to represent all possible values.

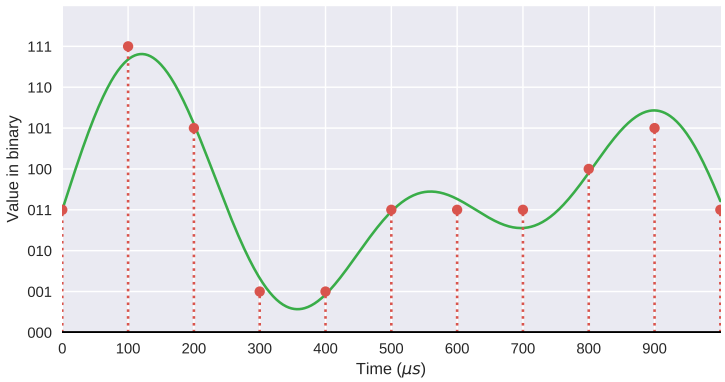


Figure 2.5: Quantization of a continuous signal. The bit depth is 3 bits and the sampling rate is 10000 Hz.

2.2.1 Frequency Domain

Our human auditory systems do not distinguish sounds by their waveforms but by the frequencies they contain over a short period of time. For any discrete signal, the maximum frequency it can represent is the Nyquist frequency, given by half the sampling frequency [25].

The frequencies a signal contains can be represented by a power spectrum. For a discrete signal $x[n]$ with N values, this can be obtained by the following expression:

$$P[f] = \frac{2}{N^2}\left|\sum_{n=0}^{N} x[n]e^{-2\pi i f n/f_s}\right|^2 \tag{2.7}$$

Where f_s is the sampling frequency and $P[f]$ the power for frequency f in Hertz up to the Nyquist frequency.

2.2.2 Mel Frequency Scale

If we look at a common audio power spectrum, we can observe that most energy seems to be confined to the lower frequencies. As such, the human auditory system evolved to perceive pitch in a logarithmically rather than in a linearly, allowing it to better differentiate frequencies in the lower registers. We can easily perceive this in musical pitches: while we perceive the difference between two notes an octave apart as constant, the rising of an octave actually represents a doubling in frequency.

The mel scale [24] is a scale that translates frequencies on hertz to how auditory systems perceive pitch. This is achieved by applying a logarithm with base two, thus assuring that the difference between octaves remains constant. This scale is normalized in such a way that it maps the values of 0 Hz and 1000 Hz to 0 mels and 1000 mels, respectively. The conversion from hertz to mels is done by the following equation:

$$m = 1000 \times log_2\left(1 + \frac{f}{1000}\right) \tag{2.8}$$

Where m is the pitch in mels and f is the frequency in hertz.

2.2.3 Audio Features

Given the first step in our machine learning algorithms return power spectrum, in the present work we chose to use features from the frequency domain, explained in more detail in the following subsections.

2.2.3.1 Logarithm of each frequency

In signal analysis and processing it is common to use decibels [19] (dB) to better represent and compare the power of sound signals. The logarithmic nature of the decibel allows better differentiation between values spread out different orders of magnitude. The formula for translating a power to decibels is described by the following equation:

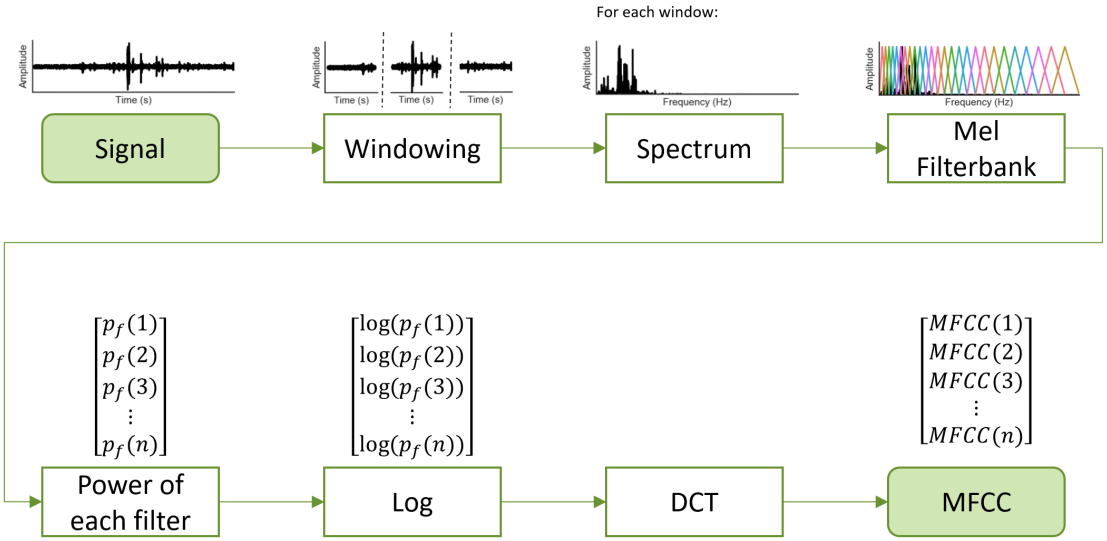


Figure 2.6: Diagram explaining how to compute MFCC.

$$P(dB) = 10 \times log(\frac{P}{P_0}) \tag{2.9}$$

Where P is the power of each frequency in the spectrum, and P_0 is a reference power. Common values for this reference include the threshold of human hearing for sound (dBA) and the value of one volt for electronically transmitted signals (dBV).

From equation 2.9, the division of the intensity by a reference value and posterior multiplication of the logarithm by 10 will only result in a shift in mean and standard deviation, respectively. Since all features are normalized and given the lack of a reference, we chose to take only the logarithm of the power for each frequency.

2.2.3.2 Mel Frequency Cepstral Coefficients

The MFCC are a group of features commonly used in speaker [14] and speech recognition [29]. These have also been used in recognition of human activities based in recorded sound [7].

To compute the MFCC from the obtained power spectrum, we start by applying a filter bank of triangular overlapping windows whose frequencies are equidistant in the mel scale. Then a discrete cosine transform is applied to the logarithms of the sum of all values in each frequency band. This process is illustrated in figure 2.6. The discrete cosine transform $X(k)$ is given by:

$$X(k) = \sum_{n=0}^{N-1} x(n) cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \tag{2.10}$$

2.2.3.3 Additional Spectral Features

Other type of common features used in audio analysis are the spectral features [20]. These features were also extracted from the background frequency spectrum. Defining $f[n]$ the frequency corresponding to the nth bin and $P[f]$ as its respective intensity, the following features were calculated:

- **Centroid**: The barycenter of the spectrum, and it is computed by the 1^{st} order moment:

$$\mu = \frac{\sum_n f[n] \cdot P[n]}{\sum_n P[n]} \tag{2.11}$$

- **Spread**: The variance of the spectrum around its centroid, and it is computed by the 2^{nd} order moment:

$$\sigma^2 = \frac{\sum_n (f[n] - \mu)^2 \cdot P[n]}{\sum_n P[n]} \tag{2.12}$$

- **Skewness**: Gives a measure of the asymmetry around the centroid, and it is computed by the 3^{rd} order moment:

$$m_3 = \frac{\sum_n (f[n] - \mu)^3 \cdot P[n]}{\sum_n P[n]} \tag{2.13}$$

- **Kurtosis**: Gives a measure of flatness of the spectrum, and it is computed by the 4^{th} order moment:

$$m_4 = \frac{\sum_n (f[n] - \mu)^4 \cdot P[n]}{\sum_n P[n]} \tag{2.14}$$

- **Slope**: Represents the amount of decreasing of the spectral amplitude, and it is computed by its linear regression:

$$slope = \frac{1}{\sum_n P[n]} \cdot \frac{N \sum_n f[n]P[n] - \sum_n f[n] \sum_n P[n]}{N \sum_n f^2[n] - (\sum_n f[n])^2} \tag{2.15}$$

- **Decrease**: Also represents the amount of decreasing of the spectral amplitude, and its formulation comes from perceptual studies, correlating to human perception:

$$decrease = \frac{1}{\sum_{n=1:N} P[n]} \cdot \sum_{n=2:N} \frac{P[k] - P[0]}{k - 1} \tag{2.16}$$

- **Roll-Off**: The frequency so that 95% of the signal's energy is contained below this frequency. This frequency must satisfy the following condition:

$$f[k] : \sum_{n=0}^{k} P[n] \geq 0.95 \cdot \sum_{n=0}^{\infty} P[n] \tag{2.17}$$

Chapter 3

Proposed Framework

3.1 SoundSignature: Indoor Location based on Background Spectrum Analysis

Every location has its own distinct set of constant background noises. These may have multiple origins, such as the humming of air conditioning systems, the noise of cars passing or a nearby road or the chirping of birds by the windows.

These background sounds are further modulated by the locations physical properties such as dimensions, materials used in its construction and placement of objects such as rugs and wooden furniture. These characteristics determine the location's impulse response, which is the result of the echoes produced by an unit pulse in said location and acts as a filter for any sound produced in it.

Given these properties, we assume that each room has acoustic characteristics that are sufficiently constant and unique for identification. The algorithm proposed in this chapter consists of a classifier makes use of these characteristics to predict the location the user is most likely to be in, from a set of previously recorded places.

The structure of the resulting algorithm can be seen in figure 3.1.

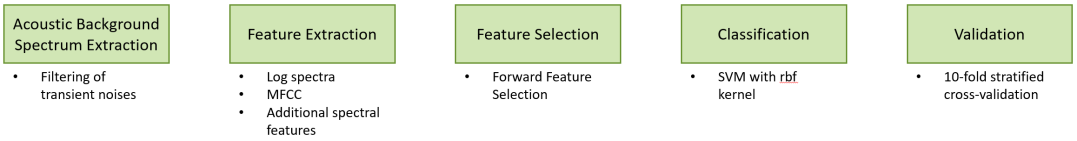


Figure 3.1: Schematic representation of the SoundSignature algorithm.

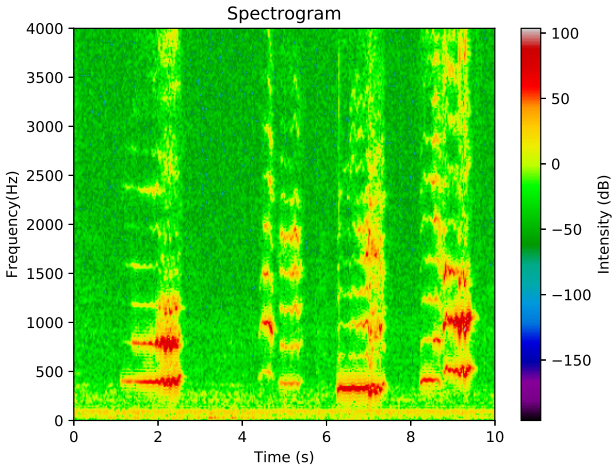


Figure 3.2: In this spectrogram of an segment of audio data we can visually discriminate two distinct components: a background noise spectrum that remains constant throughout the signal and transient sounds of larger intensity that are addictive to this spectrum.

3.1.1 Acoustic Fingerprint Extraction

Given an audio recording labeled with the location it was recorded in, we start by dividing it into non-overlapping windows of t_{win} length. Then we proceed to compute its spectrogram, which is a representation of the frequency spectrum as it varies with time. This is achieved by:

1. Divide the segment into frames of t_{frame} length;
2. Multiply each segment by a window function to reduce the signal's amplitude near the boundaries, reducing some possible high frequency artifacts;
3. Compute the power spectrum for each frame;
4. Remove the redundant second half of the result;
5. Merge the result of each frame into a bidimensional map, resulting in a representation of the signal's power for a given frequency at a given time.

As shown in the spectrogram from figure 3.2, two distinct components can be identified in the spectrogram:

- The **background noise spectrum**, that relates to the aforementioned intrinsic acoustic characteristics of the location. This spectrum remains consistent throughout the spectrogram.
- **Transient sounds**, which are the cause of limited duration events such as speech or the sound of a door closing. These are always addictive to the background noise spectrum.

To separate this background frequency from the remaining sounds we could extract the minimum of each frequency along the spectrogram and merge them together to create a spectrum. However, this method is prone to errors due to a number of reasons:

- Common smartphone's sound recording systems typically have some inherent **dynamic audio compression**, which means that the larger the input signal, the lower the gain. This leads to dips in the signal's gain following the recording of loud noises, causing readings with lower intensity than it is supposed to be.

- **Electronic noise** may also lead to recorded intensities lower than expected.

Therefore, we instead extract the 5th percentile of each frequency, which is the value below which 5% of the observations may be found. This value is chosen because it is close to the minimum while being more robust to these effects.

The result is a spectrum that only contains information related to the background noise spectrum, and thus is affected only by the aforementioned acoustics characteristics. Therefore, we can use this spectrum as an acoustic fingerprint.

3.1.2 Feature Extraction

The extracted acoustic fingerprint consists of a power spectrum, making impossible the extraction of features in both the time and statistical domain. As such, we proceed to the extraction of the features described in section 2.2.3, which are all in the frequency domain. These features consist of three groups: logarithms of the power for each frequency, MFCC and additional spectral features.

All features were standardized according to equation 2.1.

The best features are selected using the Forward Feature Selection described in section 2.1.4.

3.1.3 Classification

A classifier algorithm is an algorithm that maps input data to a category. It does so by generating a model from a training set containing observations whose category is known. It is on the basis of these that supervised learning are built upon.

There are multiple classifier algorithms. The choice of the algorithm will depend on various parameters such as the kind of data, number of observations and number of individual categories. In the present work, we chose to use SVM, as it was the most prevalent in the reviewed literature. This algorithm is described in detail in section 2.1.3.1.

When extending the algorithm to multiclass problems, there are two options: OvR and OvO. While the first approach implements a smaller number of classifiers and has

therefore better computational performance, the resulting class imbalance between positive and negative samples in each classifier usually leads to worse results. As such, in the present work we chose to use the OvO approach.

3.1.4 Validation

As explained in section 2.1.5, using the same dataset for both training and testing may lead to overly optimistic results because of overfitting. To prevent this it is common to validate classifier algorithms resorting to cross validation.

In the present work, we used a common cross-validation method: K-fold cross-validation. This method randomly splits the data into K folds and creates K classifiers. Each classifier uses a different layer for testing and the remaining K-1 layers for training. The estimated accuracy of the classifier is equal to the average of the K classifier's accuracy. Likewise, the confusion matrix can be obtained by summing all the resulting confusion matrices.

3.2 SoundSimilarity: Proximity Detection from Real-time Comparison of Audio Signals

In section 3.1, we studied how we can recognize a device's location by examining the sound perceived by its microphone. In this section, we develop a tool for further improving the proposed positioning algorithm by comparing in real time the sounds received by various users.

If the sound perceived by two users is similar, we can assume they are in the same location. Therefore, if the locations predicted by the SoundSignature algorithm for each user are different, we conclude that at least one of them is wrong. By comparing the prediction probabilities of each user we can identify which prediction is wrong and correct it.

As such, the objective of the SoundSimilarity algorithm is to develop a binary classifier that identifies whether two sounds are similar. To achieve this we developed a novel measure of similarity between the sounds recorded by two different devices. Such measurement must have the following properties:

i) Have **positive correlation with similarity** between a pair of signals, meaning that the more similar are the signals, the larger will this measurement's value be;

ii) **Immune to small time misalignments**, as it is difficult to perfectly align signals from different devices in real time;

iii) **Immune to phase differences**, as different positions in the same locations may perceive different sounds in different phases;

iv) Allow **comparisons between sounds from different devices**;

v) Given this algorithm is meant to correct possible mistakes from the SoundSignature algorithm, any sort of training on the same previously recorded data could lead to the same errors. Therefore, the proposed SoundSimilarity algorithm must be **independent of any previous knowledge**.

The structure of the resulting algorithm can be seen in figure 3.3.

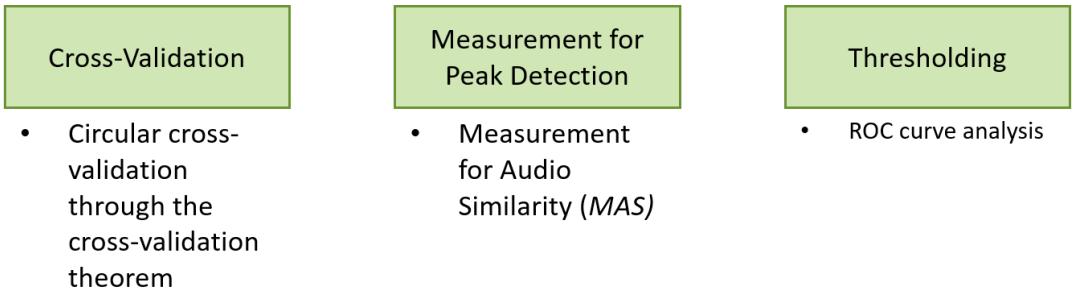


Figure 3.3: Schematic representation of the SoundSimilarity algorithm.

3.2.1 Cross-Correlation

In order to evaluate the similarity between two recorded audio signals, a cross-correlation was used. Cross-correlation can be defined as measure of similarity between two series as a function of the displacement between them. For two real valued discrete signals $f[n]$ and $g[n]$, it can be formulated as:

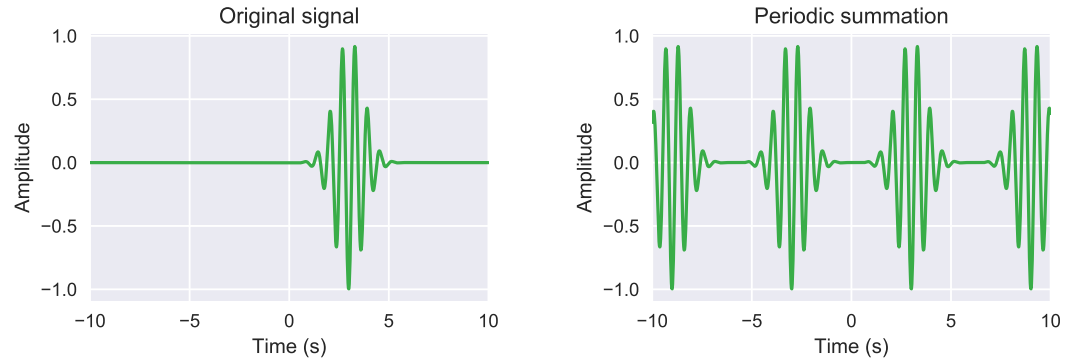
$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[m+n] \tag{3.1}$$

Given the finite length of the data, common practice is to extend the series with leading and trailing zeros. However, this method is prone to errors due to the resulting tendency to give more weight to central values, where these zeros do not interfere with the result. Therefore, we instead employ circular cross-correlation, where the input series are extended with periodic summations as in figure 3.4.

Given the discrete-time Fourier transform already employs this extension, we can employ the cross-correlation theorem and formulate the discrete circular cross-correlation as:

$$(f \star g)[n] = \mathcal{F}^{-1}\{\mathcal{F}*[f] \cdot \mathcal{F}[g]\}[n] \tag{3.2}$$

Where $\mathcal{F}$ represents the discrete-time Fourier Transform, $\mathcal{F}*$ its complex conjugate and $\mathcal{F}^{-1}$ the inverse discrete-time Fourier Transform. This formulation both avoids the aforementioned artifact and greatly improves computational performance.



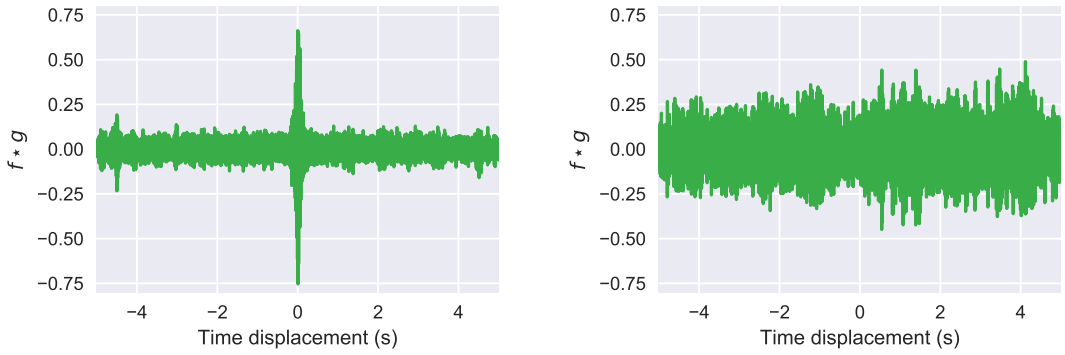
(a) Original signal.

(b) Signal with periodic summation.

Figure 3.4: Illustration of periodic summation. This process consists of taking a signal limited in time and repeating it from $-\infty$ to ∞, creating a periodic signal.

3.2.2 Measuring the Similarity of Audio Segments

Given the circular cross-correlation between two finite series, the series will be similar if a pronounced peak is present, as illustrated in figure 3.5a. As such, we propose a novel measurement for audio similarity (MAS) correlated to the presence of a peak.



(a) Signals recorded in the same location.

(b) Signals recorded in different locations.

Figure 3.5: Comparison between two cases for circular cross-correlation of audio signals recorded at the same time.

Let f and g be two series corresponding to the recordings of two microphones in a window of t_{win} length. Assume the misalignment in time between them is $t_{delay} < t_{win}/d$, where d is a free parameter. Given a circular cross-correlation $C_{f,g}$ between these two series, we start by taking the absolute value of each value in the correlation:

$$C_{abs} = |C_{f,g}| \tag{3.3}$$

Then we define a region R centered around $t_{win}/2$ and with width t_{win}/d:

$$R = [\frac{t_{win}}{2} - \frac{t_{win}}{2d}, \frac{t_{win}}{2} + \frac{t_{win}}{2d}] \tag{3.4}$$

Finally, we define the MAS as:

$$MAS_{f,g} = 1 - \min(\frac{\max_{t \in \overline{R}} C_{abs}}{\max C_{abs}}, 1) \tag{3.5}$$

If a pronounced correlation peak is present, it will be contained in the region R. Therefore, the overall maximum of the correlation will be larger than the maximum in $\overline{R}$, the region complementary to R. The greater this difference the closer to zero the ratio $\frac{\max_{t \in \overline{R}} C_{abs}}{\max C_{abs}}$ will be.

On the other hand, if no correlation peak is present, we can suppose that the correlation has the properties to those of noise. In this case, two situations can occur: if the overall maximum is contained in $\overline{R}$, we can obviously say that it is equal to the maximum in $\overline{R}$, and therefore the ratio between them will be one; if the overall maximum is contained in R, it will be approximately equal to the maximum in $\overline{R}$, and therefore the ratio between them will be approximately one. These regions are depicted in figure 3.6.

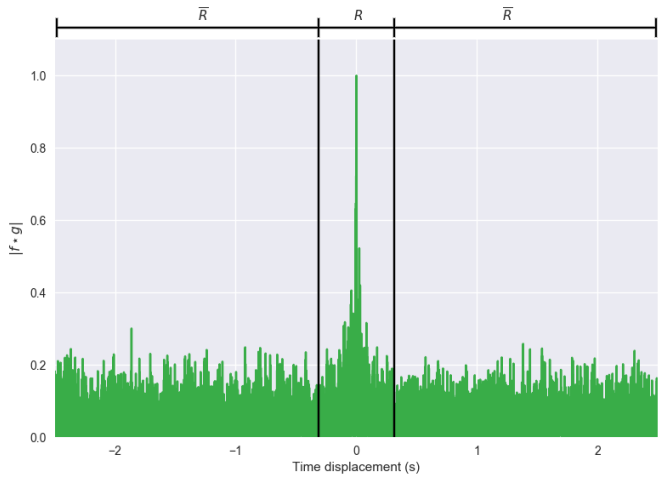
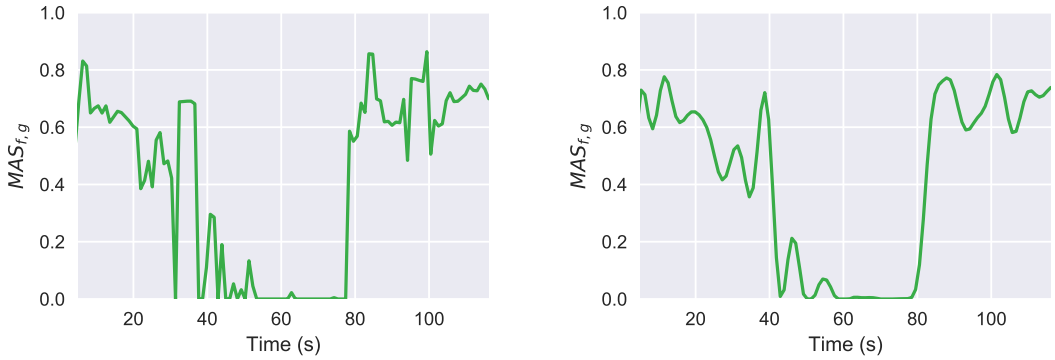


Figure 3.6: Graph of the absolute values of a correlation with a peak, normalized to $[0, 1]$. $MAS_{f,g}$ will be equal to the maximum in R minus the maximum in $\overline{R}$.

Given these properties, the result of this measurement is a value between 0 and 1 where the greater the value, the more similar are recorded sounds used as input. Furthermore, by defining a range where the correlation peak can be instead of a fixed point, we make the algorithm robust to small misalignments in time and phase shifts.

If we divide two aligned audio signals into windows of equal length and compare those that correspond to the same time interval, we can obtain a graph of audio similarity along time. While the resulting graph allows to visually determine when the compared microphones were in the same location, we can observe some high frequency artifacts. To remove these we employ a low pass filter to smooth the signal. The comparison between before and after filtering can be observed in figure 3.7.



(a) $MAS_{f,g}$ before filtering.

(b) $MAS_{f,g}$ after filtering.

Figure 3.7: Graphic of the MAS over time between two signals and comparison between before and after filtering.

3.2.3 Binary Classification

Obtained a measurement of similarity that satisfies the requirements in section 3.2, we proceed to build a binary classifier from it. The simplest method is to apply a threshold value. If the measurement of similarity is greater than this value, the sample will be classified as positive and we consider that the two compared devices are in the same location. Otherwise, the sample will be classified as negative and we consider that the two compared devices are in different locations.

3.2.3.1 Sensitivity and Specificity

When building a binary classifier from labeled data there are multiple metrics that can be extracted. Two of these are $sensitivity$ and $specificity$. $Sensitivity$ determines the probability that a labeled positive sample is classified as such and is defined by:

$$sensitivity = \frac{TP}{P} \tag{3.6}$$

Where TP is the number of positively labeled samples classified as such and P is the total number of samples labeled as positive.

Similarly, $specificity$ refers to the probability of a sample labeled as negative being classified as such. It is defined by:

$$specificity = \frac{TN}{N} \tag{3.7}$$

Where TN is the number of negatively labeled samples classified as such and N is the total number of samples labeled as negative.

3.2.3.2 Receiver Operating Characteristic curve

Varying the threshold affect both sensitivity and specificity: higher values increase specificity, while lower values increase sensitivity. To visualize the the trade-off between these effects we plot the $sensitivity$ against $1 - specificity$. This is called a ROC.

Youden's J statistic is a performance metric extracted from the ROC curve that is equal to the probability a correct classification. In the plot of a ROC curve, Youden's J statistic for a certain threshold is equal the vertical difference between the ROC and the diagonal random chance line, as represented in figure 3.8 and it is calculated by the following equation:

$$J = sensitivity + specificity - 1 \tag{3.8}$$

We choose the threshold that maximizes Youden's J statistic. If a measurement is above this threshold, the devices are in the same location. If a measurement below this threshold, the devices are in different locations.

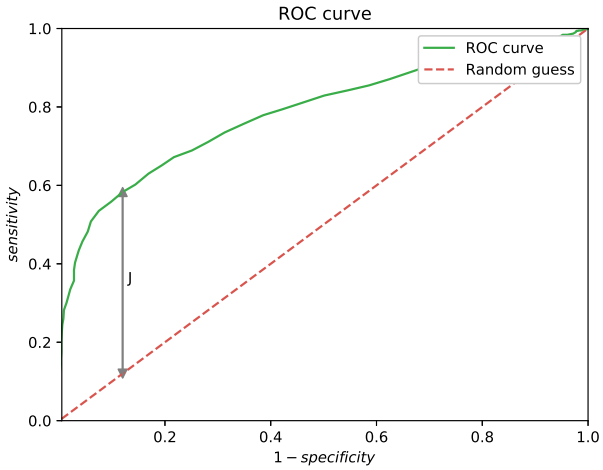


Figure 3.8: ROC curve depicting Youden's J statistic.

Another performance metric that can be extracted from the ROC curve is the Area Under the Curve (AUC). This metric is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming "positive"ranks higher than "negative"). It varies between 0.5 and 1.0 and is used when determining if a measurement is suitable for a certain binary classification problem (the greater the value, the more suitable is the measurement).

3.3 Activity Monitoring

Similarly to the SoundSignature algorithm described in section 3.1, the objective of the algorithm described in this section is to learn from labeled audio data how to classify

future inputs. However, instead of identifying the location the user is in, this algorithm recognizes everyday activities such as brushing teeth, talking and watching television. As such, the proposed algorithm is focused in the field of HAR.

Many activities are location dependent. Brushing teeth, for example, is an activity typically done in a bathroom, and the capturing the sound of a television indicates proximity to said object. This allows the algorithm to be used in conjunction to the other two previously described algorithms for indoor location.

Given that the SoundSignature algorithm already implements a sound-based machine learning architecture, we chose to reuse it in the context of HAR. The only difference in its implementation lies in the first step, the acoustic fingerprint extraction explained in section 3.1.1.

The structure of the resulting algorithm can be seen in figure 3.9.

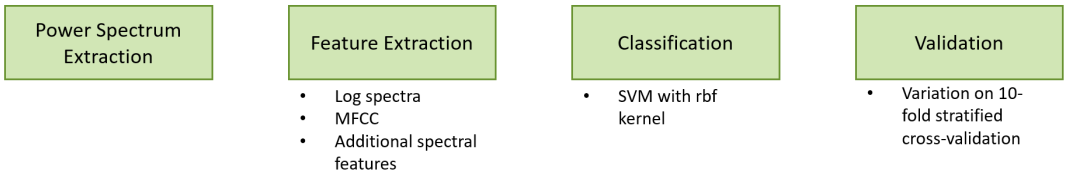


Figure 3.9: Schematic representation of the activity monitoring algorithm.

3.3.1 Preprocessing

In the case of identifying locations, the first step was to isolate the background noise from sounds of limited duration. However, in many cases the sound derived from an activity has exactly this property of being limited in time, and as such by applying this fingerprint extraction the information relative to the activity would be lost. Because of this, only the power spectrum is extracted from each window. This permits the segmentation of the signal into shorter window, allowing the extraction of more samples from our data. The signal was therefore split into windows of 0.1 seconds from which the power spectra are extracted.

3.3.2 Feature Extraction

As in the SoundSignature algorithm, three groups of features were considered: the logarithm of the power of spectra, MFCC and additional spectral features.

The increase in the amount of samples made the implemented feature selection algorithm unusable due to the computational power needed. As such, the groups of features were tested separately.

3.3.3 Classification

As in the SoundSignature algorithm, the chosen classifier algorithm was SVM. The choice was made based on its prevalence in the reviewed literature.

To compensate for the loss in computational performance derived from the increase in samples, the chosen strategy for extending the algorithm to a multiclass problem was OvR.

3.3.4 Validation

When extracting multiple samples from the same audio recording, it is expected that these will be more similar between them than samples from different recordings. Because of this, a variation on the 10-fold stratified cross-validation was implemented:

1. Split the audio recordings into 10 groups, each with equal amount of recordings of each class;
2. In each group, segment the recordings into samples of 0.1 seconds and extract the desired features;
3. Use the resulting groups as folds for the regular 10-fold cross-validation algorithm.

Chapter 4

Results

To validate the proposed algorithms, separate datasets were recorded or downloaded from the Internet.

4.1 SoundSignature

4.1.1 Proof of Concept

In a first approach, a small dataset was recorded to function as a simple proof of concept. This dataset consisted of several audio recordings where the recording device would just stay in the same place for 60 seconds, either in the hand of the user or laying in a flat surface. The locations chosen consisted of physically distinct rooms such as an office or a balcony.

Each recording was labeled with a tag indicating the place it was recorded in and split into 5 second windows to be used with the previously described algorithm. Table 4.1 shows the label of each recorded place and the number of 5 second windows recorded.

Table 4.1: Composition of the dataset used for proof of concept.

Location	Number of windows
office	31
hallway	30
entryhall	20
restaurant	17
livingroom	9
openspace	8
restaurant_balcony	3
Total	**125**

The recording device used was an iPhone 6, and the used sampling rate was 22050 Hz. Applying the algorithm described in section 3.1 returns an accuracy of 92.1%. Figure 4.1 shows the resulting normalized confusion matrix.

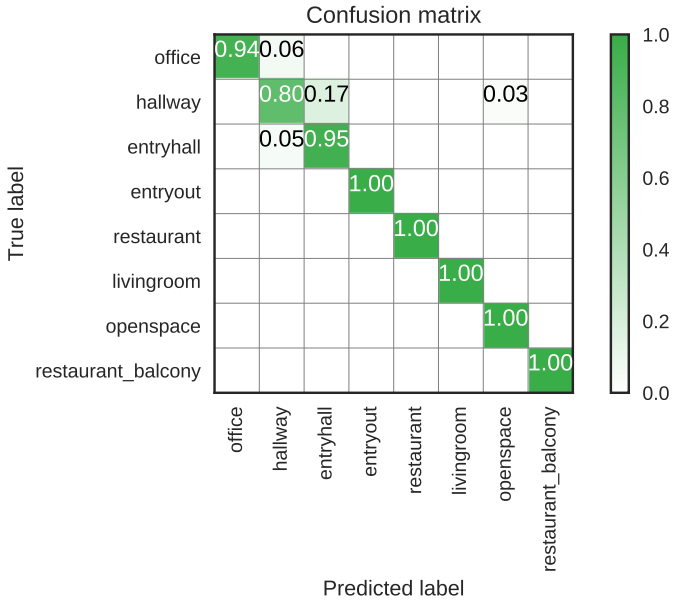


Figure 4.1: Normalized confusion matrix of the dataset used for proof of concept.

4.1.2 Data Acquisition

Achieved a proof of concept, a larger dataset named SoundSignature dataset was recorded for further validation. A series of routes for data acquisition were designed, where each route consists of a series of numbered checkpoints along a path. The user has to clear these checkpoints in a determined order while holding a smartphone in hand positioned as if they were texting. Nine routes were designed, and they are represented in appendix A.

The division of the audio recordings was done in two different splitting methods, each with its own distinct purpose:

- **Within room localization:** Each segment corresponded instead to the sound recorded in the interval between two checkpoints. For example if route *A* has five checkpoints, we obtain four segments and the first one is labeled as *A0:A1*.
- **Distinction between rooms:** Segmentation was performed as in the previous dataset by splitting each recording into segments of 5 seconds each and labeling them according to the location they were recorded in.

The composition of this dataset is show in table 4.2.

Table 4.2: Composition of the SoundSignature dataset.

Route	Location	Number of checkpoints	Number of recordings	Number of 5 second windows
A	hallway	5	17	99
B	hallway	5	15	88
C	hallway	3	15	40
D	office1	5	16	47
E	entryhall	8	10	71
F	entryhall	5	10	72
G	office2	4	10	25
H	bathroomUp	2	12	56
I	bathroomDown	2	12	56
Total		111	117	579

All recordings were acquired using Recorder, an Android app developed by Fraunhofer AICOS. This app allows recording multiple sensors simultaneously and provides a tap counter for marking events such as passing through checkpoints. The folder structure of the recordings is determined by their metadata (such as time of recording, name of the route and phone used), providing both organization for the data and easy access to this information. The recording device used was a LG Nexus 5 smartphone.

The application records at a sampling rate of 8000 Hz, achieving a maximum discernible frequency or Nyquist's frequency is 4000 Hz. Each frame in the computation of the spectrogram for the acoustic fingerprint extraction consists of 512 samples, resulting in 257 frequency bins per frame. As a result, each acoustic fingerprint consists of 257 bins, each corresponding to a frequency from 0 Hz to 4000 Hz.

As for the number of features, the logarithms of the power of each frequency create 257 features. The number of filter banks used for MFCC was 26, since this number is typically used in speech related applications [8]. Along with the additional 7 spectral features, the total number of features is 290.

4.1.3 Results

Using the splitting method for within room localization, the SoundSignature algorithm shows poor results, yielding an accuracy of 22.45%. Figure 4.2 shows the resulting normalized confusion matrix. The best subset of features found by the feature selection algorithm were the log powers of 46.875Hz and 78.125 Hz, along with MFCC's number 11 and 15.

If we instead choose as labels the locations where each route was recorded, we obtain a result more similar to the one obtained in the proof of concept. Applying the algorithm with such labels obtains an accuracy of 86.48%, and figure 4.3 shows resulting normalized confusion matrix. The feature selection algorithm reduced the number of features from 290 to 10. All features selected were from the first group, the logarithms for each

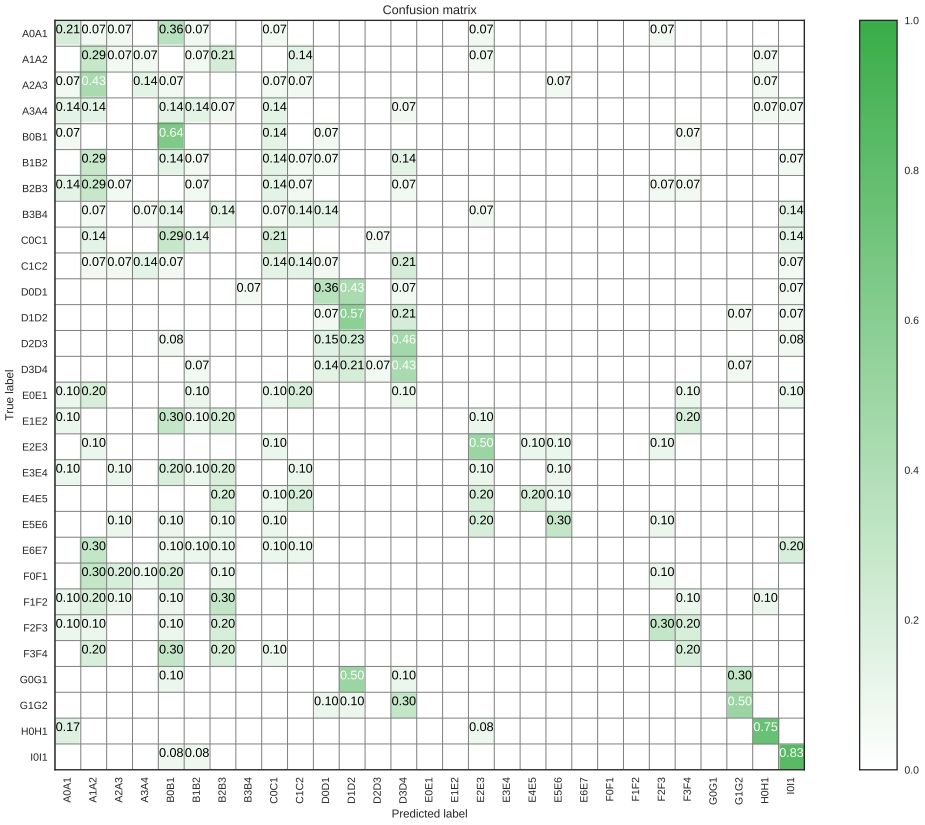


Figure 4.2: Normalized confusion matrix of the SoundSignature algorithm with the SoundSignature for classification within the locations.

frequency bin. In order of importance, the selected features were 1265.625 Hz, 296.875 Hz, 1796.875 Hz, 156.25 Hz, 109.375 Hz, 1515.625 Hz, 3953.125 Hz, 250.0 Hz, 375.0 Hz and 343.75 Hz.

The normalized confusion matrix shows that the greatest confusion was between the *entryhall* and the *hallway* locations. By consulting which routes were used for each location in 4.2 and visualizing said routes in appendix A, we can observe that not only are these locations adjacent to each other but also they are the only not physically separated by neither doors or walls.

For further validation of the algorithm, a test route was designed for simulating the use of the trained algorithm in real time. This route passed through each previously recorded location. The recorded sound was then split into 5 second windows which were then classified by the algorithm. This test resulted in an accuracy of 83.40%, and its comparison to the ground truth can be observed in figure 4.4.

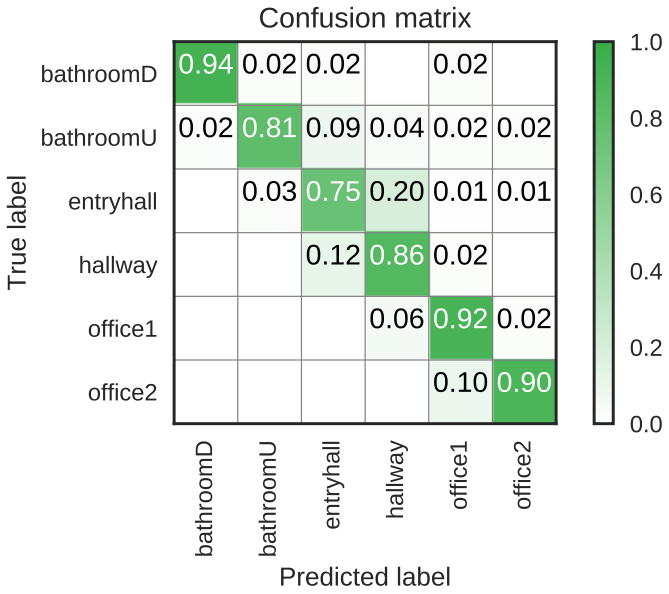


Figure 4.3: Normalized confusion matrix of the SoundSignature algorithm with the SoundSignature for differentiating locations.

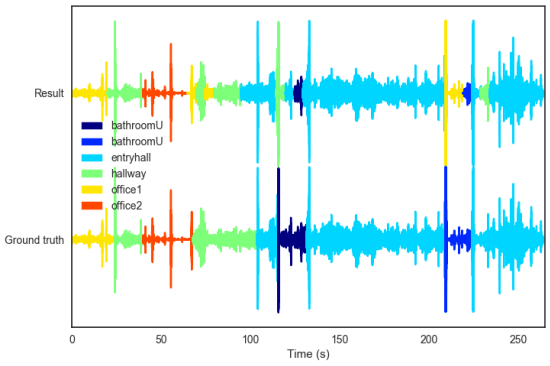


Figure 4.4: Result of the classification of the test route. Below is the ground truth; above is the result of the classification.

4.2 SoundSimilarity

4.2.1 Data Acquisition

For each acquisition made for designing and testing the SoundSimilarity algorithm used two sound recording devices. A variety of devices was used, namely an Apple iPhone 6, an Apple iPad Air and an HP Envy laptop computer. One of the devices was left stationary in a predetermined position, while the other was taken by the user held in texting position while they walked through a predetermined route.

The routes were designed in such a way that initially the devices are in the same location. The user then leaves this location and then returns. The times of leaving and

reentry are annotated. figure 4.5 shows a graphical representation of such an acquisition.

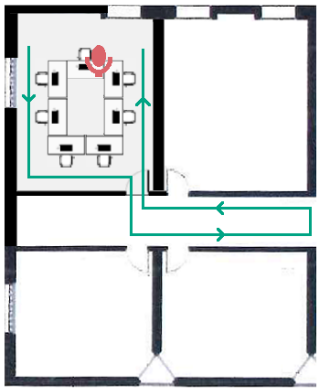


Figure 4.5: Illustration of a path designed for data acquisition for the SoundSimilarity algorithm. The red microphone symbol represents the stationary microphone, the blue circle represents the starting position and the green line represents the path the user takes while holding a sound recording device.

Eleven such acquisitions were made. Although different devices recorded at different sampling rates, all signals were downsampled to 8000 Hz, increasing computational performance and ensuring all signals have the same sampling rate.

4.2.2 Data Processing

Each acquisition was preprocessed independently. To simulate usage of the algorithm in real time, the first step was to align the signals from both microphones. Since that at the start both the subject and the fixed microphone will be at the same location, we can assume that a peak will be present in the correlation between the first seconds. As such, for two signals f and g, a correlation between the first $t'_{win} = 10s$ of each signal is computed. The misalignment t_{delay} is given by the following expression:

$$_{delay} = \frac{t'_{win}}{2} - t[argmax(f \star g)] \tag{4.1}$$

Both recordings were then split into windows of five seconds with an overlap of four seconds, resulting with a pair of aligned five second windows every second. For each pair the MAS is computed as described in section 3.2.2. The resulting graph of sound similarity over time is then filtered with a low-pass of order 5 and cut-off frequency of 0.3 Hz. The result can be seen in figure 3.7b.

4.2.3 ROC curve analysis

Each point in all eleven computed similarity over time graphs is labeled according to whether the devices were in the same location (positive label) or not (negative label). This allows the construction of a ROC curve, shown in figure 4.6.

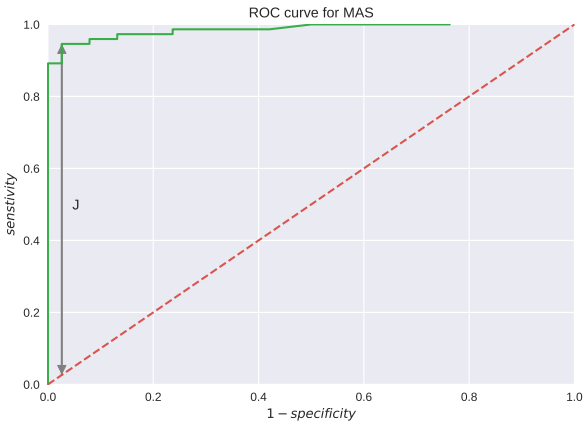


Figure 4.6: ROC curve for the MAS.

The AUC of this graph is 0.983. This means that if we randomly select a positive and a negative instance, there is a 98.3% probability of that the MAS will be higher for the positive instance than for the negative one.

By maximizing the Youden's J statistic as described in section 3.2.3.2 we obtain a optimal threshold of 0.261.

4.2.4 Results

Values above the determined threshold will be classified as positive, meaning the devices whose sound is compared are in the same location. On the other hand, values below this threshold will be classified as negative, meaning the devices are in different locations.

This result is contrasted against a ground truth in figure 4.7. Among the recorded samples, the algorithm could identify if the devices were in the same room 94.59%. In the shown example of figure 4.7 these errors occurred only in the transition between areas, where it may be difficult to even establish a ground truth.

4.3 Activity Monitoring

4.3.1 Dataset

The dataset used was downloaded from the link `sounds.natix.org` and was recorded by the authors of the article [1]. The portion of the dataset used consisted of multiple labeled recordings of various common household activities and items, such as brushing teeth and the sound of a washing machine. Table 4.3 shows the composition of this dataset.

Each recording was split into 0.1 second segments. From each segment, a power spectrum was computed, from which the logarithm of the power of each frequency, MFCC and additional spectral features were extracted.

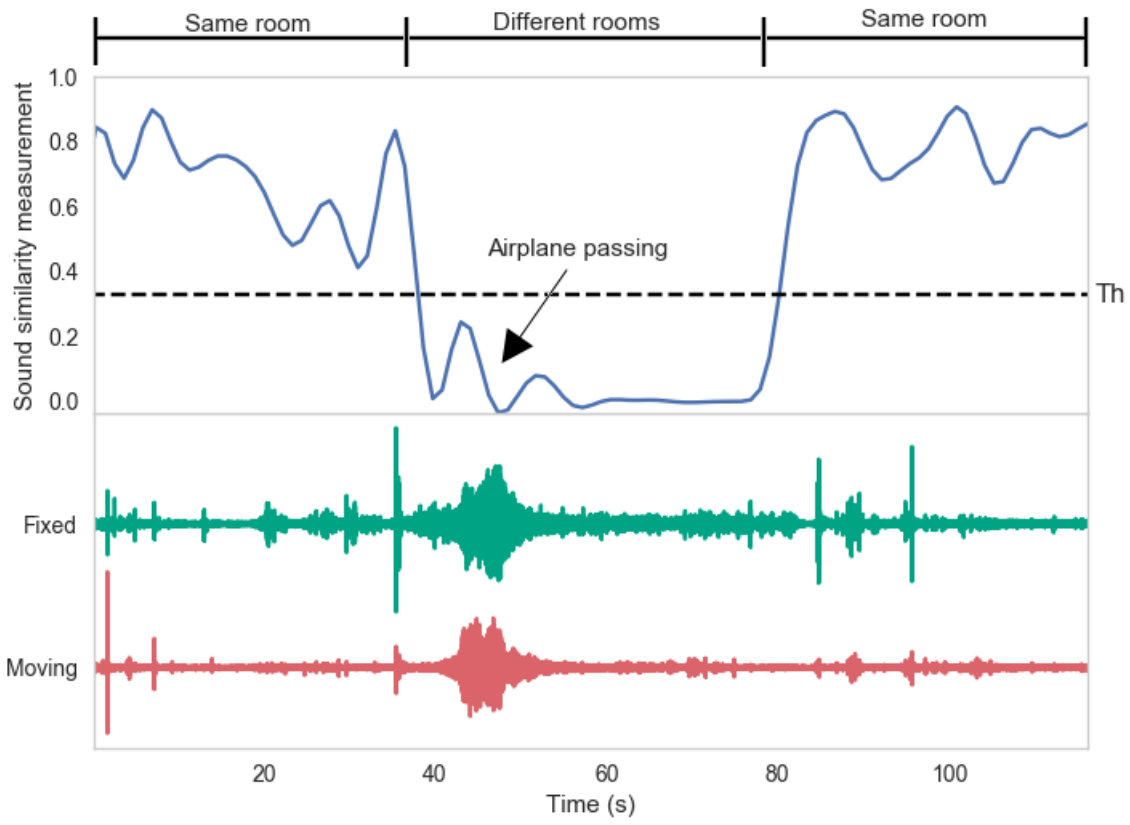


Figure 4.7: Result of the SoundSimilarity algorithm, compared to the ground truth. In the represented example an airplane redying for landing passed over the building while the two devices were in different locations, meaning that both recorded its characteristic sound. This means that both recorded the same sound, possibly generating wrong results. However, the MAS not only showed itself resilient to this event but also the threshold adapted to this scenario.

4.3.2 Results

The large amount of samples (20237) made it infeasible to apply the feature selection algorithm due to its computational requirements. As such, the three groups of features were tested individually. Applying the validation method described in section 3.3.4 with the logarithms of the powers of each frequency, MFCC and the additional features yielded accuracies of 83.16%, 97.88% and 70.50%, respectively. Figure 4.8 shows the normalized confusion matrix achieved with MFCC, the group that yielded the best accuracy.

Table 4.3: Composition of the dataset used activity recognition.

Label	Number of recordings	Number of 0.1 second windows
cough	40	1359
dyerMachine	40	1200
blender	40	1220
stew	40	1220
fondoSilencio	40	1200
flushing	40	1553
fondoTranquilo	40	1200
boilingWater	40	1200
vaccum	40	1200
tel	40	497
bike	40	1200
washingDishes	40	1200
knockingDoor	40	1200
toothBrushing	40	1200
washingMachine	40	1212
doorBell	40	1175
shower	40	1201
Total	**680**	**20237**

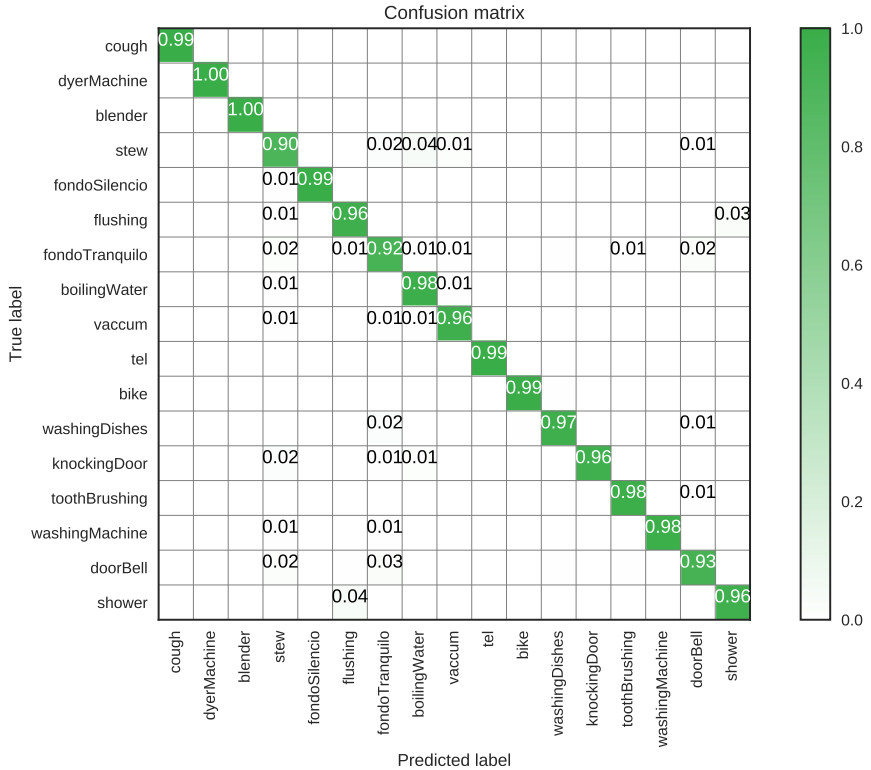


Figure 4.8: Normalized confusion matrix for the Activity Recognition algorithm.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presented a framework for locating the user and recognizing the activities they perform. It does so by analyzing the pervasive sound perceived through a smartphone's microphone.

In this paper, we presented two different algorithms for indoor localization based on sound. With these we were able to locate users and calibrate their locations by comparing the signals by them perceived, relying exclusively on pervasive sound while requiring no infrastructure. For the elaboration of these algorithms, data collection protocols were designed and followed in order to create suitable datasets.

With first algorithm, SoundSignature, we identified the location the user is in. This algorithm can be divided into two stages: an offline stage and an online stage. During the offline stage, acoustic fingerprints are extracted from the training data by filtering out transient sounds from the power spectrum of the background noise. From these fingerprints a large group of features is extracted, from which the best subset is identified through the use of a feature selection algorithm. Finally, these features are used to train an SVM classifier. During the online stage, the previously selected features are extracted from real-time audio to predict the location of the user.

Through 10-fold stratified cross validation we achieved an accuracy of 86.48%. The feature selection step reduced the number of features from 290 to 10, improving both accuracy computational performance. All features selected were logarithms of powers extracted directly from the acoustic fingerprint, and all except for one corresponded to frequencies bellow 2000 Hz. This indicates that it is possible that we can reduce the sampling frequency to 4000 Hz without much impact on the results, allowing us to further improve the algorithm's computational performance. Validation in a test route

designed to simulate a real usage environment yielded a similar accuracy of 83.40%, most errors occurring in transitions between locations. We consider that this happens due to the extraction of a 5 second window, as some contain information from two distinct locations.

While sound-based indoor location has been achieved with good results by Stephen P. Tarzia *et al.* [27], the proposed algorithm shows some significant improvements over the current state of the art:

- The proposed algorithm's data acquisition protocol does not require dedicated recording equipment, only a smartphone placed on the user while they walk through a predetermined route. This decision was led by the fact that we consider these the conditions under which an indoor positioning system would be used;
- The length of the window used was 5 seconds instead of 30, allowing a swift feedback to the user and response to location changes, a requirement for real-time indoor navigation systems;
- The data recorded for the the development of this algorithm was acquired over the span of several weeks instead of in a few days, thus granting more variability to the dataset and providing more realistic results less prone to overfitting. Despite this fact, the proposed algorithm shows an accuracy of 83.40%, a clear improvement over the 71.7% reported by Stephen P. Tarzia et al. [27].

With the second algorithm, SoundSimilarity, we detect if a user is in the same location as another user or a device. With this goal, we created a novel measure of sound similarity measurement that is based on the presence a correlation peak. This measurement's associated area under the ROC curve of 0.98 validates its use for the proposed function. By maximizing Youden's statistic in this curve we obtain an optimum threshold for the measurement. While the achieved accuracy of 94.59% is not perfect, observation of the graphs of sound similarity over time indicate that most errors moments of transition. We believe this delay is due to the use of the t_{win} second window and to the low pass filtering.

Other similarity metrics were considered, namely covariance, correlation coefficient and mutual information score, all in both time and frequency domains. Using the correlation coefficient in the frequency domain, yielded an area under the curve ROC score of 0.8803, while the remaining metrics showed AUROC scores between 0.45 and 0.55 and were as such considered unfit for this application.

Most notably, we compared the obtained results to the results obtained by Satoh *et al.* [23]. While the algorithms performed similarly in static conditions, our method showed imperviousness to perturbations such as the sound of footsteps and the passing of airplanes, while the compared method behaved poorly under these conditions. The AUROC score obtained by applying the method proposed by Satoh *et al.* to our dataset is 0.8736.

With the third algorithm, Activity Monitoring, we recognize the activities the user performs. The algorithm splits the audio recordings into windows of 0.1 seconds and for each computes a power spectrum from which relevant features are extracted. These are then used with an SVM classifier. By training and testing the algorithm with a dataset downloaded from `sounds.natix.org`, the best results were achieved with MFCC, yielding an accuracy of 97.88%. When compared to the results shown in the literature review in section 1.3 the proposed algorithm shows a clear improvement over the current state of the art. Moreover, this algorithm being a derivation of the SoundSignature algorithm may indicate that this approach can be easily adapted to other sound-based machine learning tasks, such as identifying malfunctions in assembly lines based on the sounds it emits.

5.2 Future work

Although the developed work shows promising results, we have yet to integrate these into a single indoor positioning system that identifies the location the user is in and corrects possible mistakes based on the proximity between users and the recognition of location-specific activities.

Validation with a larger dataset containing more locations and recording devices is planned. In this acquisition we will also study how different device positions such as in call position or in pocket may affect the results. The impact of reducing the sampling rate will also be studied.

Moreover, we may also study the effects of imposing limitations on transitions between locations, thus excluding physically those that are impossible. Furthermore, to minimize the errors in moments of transitions between locations, we could implement a rejection class by thresholding the degree of confidence returned by the classifier.

Finally, a mobile application will be developed to streamline the processes of data acquisition and training and validation of the developed algorithms, by allowing these to be tested in real-time.

Bibliography

[1] J. Beltran, E. Chávez, and J. Favela. "Scalable Identification of Mixed Environmental Sounds, Recorded from Heterogeneous Sources." In: 68 (Sept. 2015).

[2] J. Bordoy, A. Traub-Ens, A. Sadr, J. Wendeberg, F. Höflinger, C. Schindelhauer, and L. Reindl. "Bank of Kalman Filters in Closed-Loop for Robust Localization Using Unsynchronized Beacons." In: *IEEE Sensors Journal* 16.19 (2016), pp. 7142–7149. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2597967.

[3] L. Chen, K. Yang, and X. Wang. "Robust Cooperative Wi-Fi Fingerprint-Based Indoor Localization." In: *IEEE Internet of Things Journal* 3.6 (2016), pp. 1406–1417. ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2609405.

[4] T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, B. Harrison, B. Hemingway, J. Hightower, P. Klasnja, K. Koscher, A. LaMarca, J. A. Landay, L. LeGrand, J. Lester, A. Rahimi, A. Rea, and D. Wyatt. "The Mobile Sensing Platform: An Embedded Activity Recognition System." In: *IEEE Pervasive Computing* 7.2 (2008), pp. 32–41. ISSN: 1536-1268. DOI: 10.1109/MPRV.2008.39.

[5] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. "Miscellaneous Clustering Methods." In: *Cluster Analysis*. John Wiley & Sons, Ltd, 2011, pp. 215–255. ISBN: 9780470977811. DOI: 10.1002/9780470977811.ch8. URL: http://dx.doi.org/10.1002/9780470977811.ch8.

[6] R. Faragher and R. Harle. "Location Fingerprinting With Bluetooth Low Energy Beacons." In: *IEEE Journal on Selected Areas in Communications* 33.11 (2015), pp. 2418–2428. ISSN: 0733-8716. DOI: 10.1109/JSAC.2015.2430281.

[7] E. Garcia-Ceja, C. E. Galván-Tejada, and R. Brena. "Multi-view stacking for activity recognition with sound and accelerometer data." In: *Information Fusion* 40 (2018), pp. 45 –56. ISSN: 1566-2535. DOI: http://dx.doi.org/10.1016/j.inffus.2017.06.004. URL: http://www.sciencedirect.com/science/article/pii/S1566253516301932.

[8] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." In: *Neural Networks* 18.5 (2005). IJCNN 2005, pp. 602 –610. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2005.06.042. URL: http://www.sciencedirect.com/science/article/pii/S0893608005001206.

[9] V. Guimarães, L. Castro, S. Carneiro, M. Monteiro, T. Rocha, M. Barandas, J. Machado, M. Vasconcelos, H. Gamboa, and D. Elias. "A motion tracking solution for indoor localization using smartphones." In: *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE. 2016, pp. 1–8.

[10] S. Hijikata, K. Terabayashi, and K. Umeda. "A simple indoor self-localization system using infrared LEDs." In: *2009 Sixth International Conference on Networked Sensing Systems (INSS)*. 2009, pp. 1–7. DOI: 10.1109/INSS.2009.5409955.

[11] A. Jain and D. Zongker. "Feature selection: evaluation, application, and small sample performance." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.2 (1997), pp. 153–158. ISSN: 0162-8828. DOI: 10.1109/34.574797.

[12] J. Jo, H. Yoo, and I. C. Park. "Energy-Efficient Floating-Point MFCC Extraction Architecture for Speech Recognition Systems." In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24.2 (2016), pp. 754–758. ISSN: 1063-8210. DOI: 10.1109/TVLSI.2015.2413454.

[13] D. Kelly and B. Caulfield. "Pervasive Sound Sensing: A Weakly Supervised Training Approach." In: *IEEE Transactions on Cybernetics* 46.1 (2016), pp. 123–135. ISSN: 2168-2267. DOI: 10.1109/TCYB.2015.2396291.

[14] T. Kinnunen, R. Saeidi, F. Sedlak, K. A. Lee, J. Sandberg, M. Hansson-Sandsten, and H. Li. "Low-Variance Multitaper MFCC Features: A Case Study in Robust Speaker Verification." In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.7 (2012), pp. 1990–2001. ISSN: 1558-7916. DOI: 10.1109/TASL.2012.2191960.

[15] B. G. Lee and W. Y. Chung. "Multitarget Three-Dimensional Indoor Navigation on a PDA in a Wireless Sensor Network." In: *IEEE Sensors Journal* 11.3 (2011), pp. 799–807. ISSN: 1530-437X. DOI: 10.1109/JSEN.2010.2076802.

[16] J. Liu, Y. Chen, A. Jaakkola, T. Hakala, J. Hyyppä, L. Chen, J. Tang, R. Chen, and H. Hyyppä. "The uses of ambient light for ubiquitous positioning." In: *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*. 2014, pp. 102–108. DOI: 10.1109/PLANS.2014.6851363.

[17] Y. Ma, Z. Dou, Q. Jiang, and Z. Hou. "Basmag: An Optimized HMM-Based Localization System Using Backward Sequences Matching Algorithm Exploiting Geomagnetic Information." In: *IEEE Sensors Journal* 16.20 (2016), pp. 7472–7482. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2600099.

[18] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro, and C. J. Bleakley. "Indoor Localization Using 802.11 Time Differences of Arrival." In: *IEEE Transactions on Instrumentation and Measurement* 65.3 (2016), pp. 614–623. ISSN: 0018-9456. DOI: 10.1109/TIM.2015.2506239.

[19] W. H. Martin. "Decibel - The name for the transmission unit." In: *The Bell System Technical Journal* 8.1 (1929), pp. 1–2. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1929.tb02302.x.

[20] G. Peeters and X. Rodet. *A large set of audio feature for sound description (similarity and classification) in the CUIDADO project*. Tech. rep. Ircam, Analysis/Synthesis Team, 1 pl. Igor Stravinsky, 75004 Paris, France, 2004.

[21] S. Pergoloni, Z. Mohamadi, A. M. Vegni, Z. Ghassemlooy, and M. Biagi. "Metameric Indoor Localization Schemes Using Visible Lights." In: *Journal of Lightwave Technology* 35.14 (2017), pp. 2933–2942. ISSN: 0733-8724. DOI: 10.1109/JLT.2017.2706527.

[22] J. W. Qiu and Y. C. Tseng. "M2M Encountering: Collaborative Localization via Instant Inter-Particle Filter Data Fusion." In: *IEEE Sensors Journal* 16.14 (2016), pp. 5715–5724. ISSN: 1530-437X. DOI: 10.1109/JSEN.2016.2566679.

[23] H. Satoh, M. Suzuki, Y. Tahiro, and H. Morikawa. "Ambient Sound-based Proximity Detection with Smartphones." In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. SenSys '13. Roma, Italy: ACM, 2013, 58:1–58:2. ISBN: 978-1-4503-2027-6. DOI: 10.1145/2517351.2517436. URL: http://doi.acm.org/10.1145/2517351.2517436.

[24] S. S. Stevens, J. Volkmann, and E. B. Newman. "A Scale for the Measurement of the Psychological Magnitude Pitch." In: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190. DOI: 10.1121/1.1915893. eprint: http://dx.doi.org/10.1121/1.1915893. URL: http://dx.doi.org/10.1121/1.1915893.

[25] H. Stiltz. *Aerospace telemetry*. Prentice-Hall space technology series vol. 1. Prentice-Hall, 1961. URL: https://books.google.pt/books?id=cro8AAAAIAAJ.

[26] J. Stork, L. Spinello, J. Silva, and K. O. Arras. "Audio-Based Human Activity Recognition Using Non-Markovian Ensemble Voting." In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'12)*. 2012.

[27] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. "Indoor Localization Without Infrastructure Using the Acoustic Background Spectrum." In: *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. MobiSys '11. Bethesda, Maryland, USA: ACM, 2011, pp. 155–168. ISBN: 978-1-4503-0643-0. DOI: 10.1145/1999995.2000011. URL: http://doi.acm.org/10.1145/1999995.2000011.

[28] V. Tiwari. "MFCC and its applications in speaker recognition." In: 1 (Jan. 2010).

[29] S. Umesh and R. Sinha. "A Study of Filter Bank Smoothing in MFCC Features for Recognition of Children's Speech." In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.8 (2007), pp. 2418–2430. ISSN: 1558-7916. DOI: 10.1109/TASL.2007.906194.

[30] J. C. Wang, Y. S. Lee, C. H. Lin, E. Siahaan, and C. H. Yang. "Robust Environmental Sound Recognition With Fast Noise Suppression for Home Automation." In: *IEEE Transactions on Automation Science and Engineering* 12.4 (2015), pp. 1235–1242. ISSN: 1545-5955. DOI: 10.1109/TASE.2015.2470119.

[31] W. Xue, W. Qiu, X. Hua, and K. Yu. "Improved Wi-Fi RSSI Measurement for Indoor Localization." In: *IEEE Sensors Journal* 17.7 (2017), pp. 2224–2230. ISSN: 1530-437X. DOI: 10.1109/JSEN.2017.2660522.

[32] Y. Yang, B. Guo, Z. Yu, and H. He. *Social Activity Recognition and Recommendation Based on Mobile Sound Sensing*. Dec. 2013.

[33] W. J. Youden. "Index for rating diagnostic tests." In: *Cancer* 3.1 (1950), pp. 32–35. ISSN: 1097-0142. DOI: 10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3. URL: http://dx.doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

[34] Y. Zhan, S. Miura, J. Nishimura, and T. Kuroda. "Human Activity Recognition from Environmental Background Sounds for Wireless Sensor Networks." In: *2007 IEEE International Conference on Networking, Sensing and Control*. 2007, pp. 307–312. DOI: 10.1109/ICNSC.2007.372796.

[35] Y. Zhan and T. Kuroda. "Wearable sensor-based human activity recognition from environmental background sounds." In: *Journal of Ambient Intelligence and Humanized Computing* 5.1 (2014), pp. 77–89. ISSN: 1868-5145. DOI: 10.1007/s12652-012-0122-2. URL: https://doi.org/10.1007/s12652-012-0122-2.

Appendix

Paths for the SoundSignature Dataset

Figures A.1 and A.2 show the paths designed to acquire data for the SoundSignature dataset.

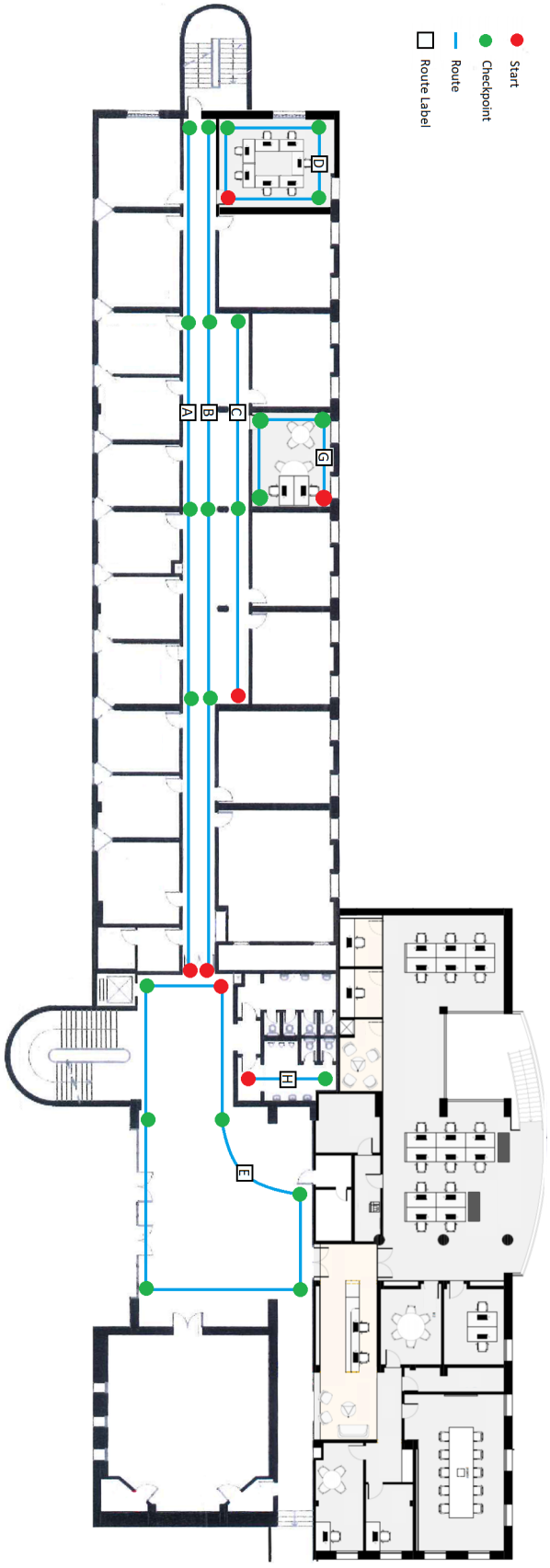


Figure A.1: Plant of the ground floor of the building where the acquisitions for the SoundSignature dataset where made. In this plant we can see the designed routes and their respective labels.

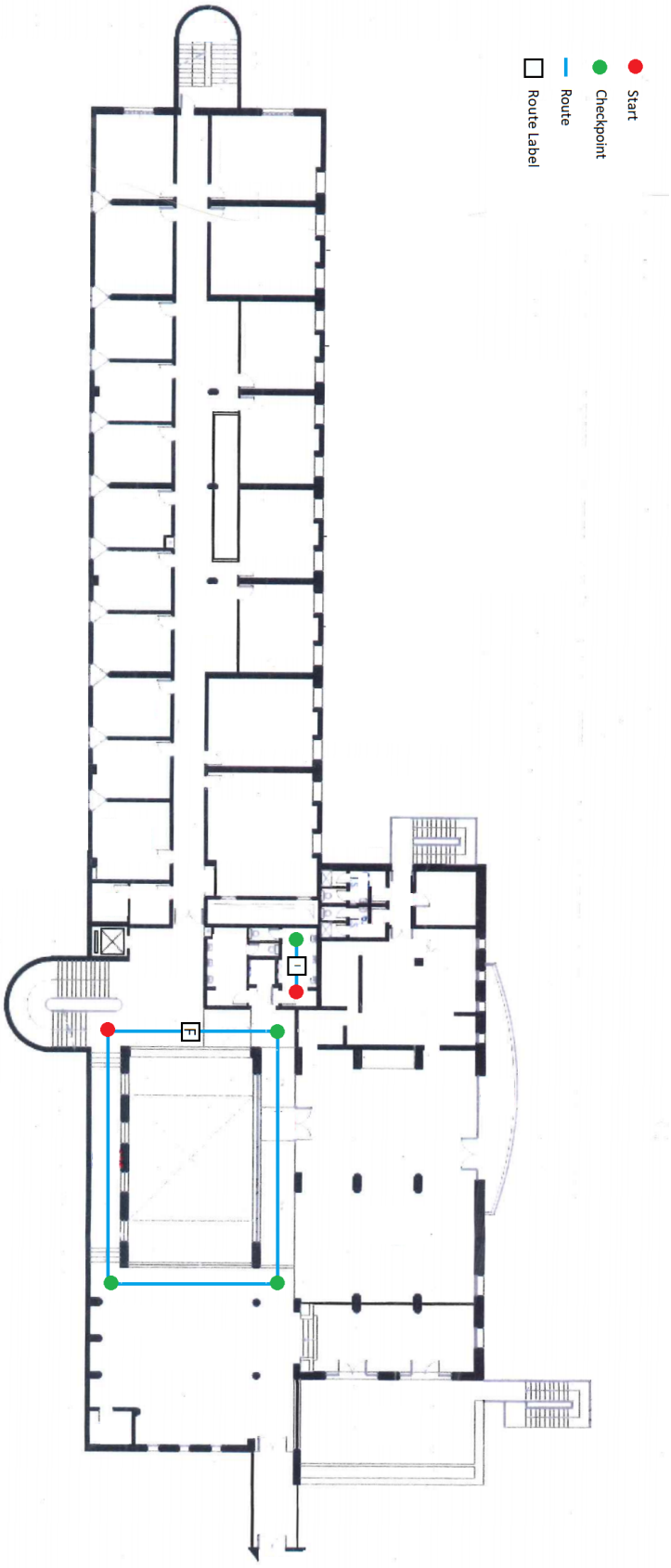


Figure A.2: Plant of the first floor of the building where the acquisitions for the SoundSignature dataset where made. In this plant we can see the designed routes and their respective labels.