Department of Informatics
University of Fribourg (Switzerland)

GENERATIVE MODELS FOR TIME SERIES IN THE CONTEXT OF IN-HOME MONITORING

THESIS

presented to the Faculty of Science of the University of Fribourg (Switzerland)

in consideration for the award of the academic grade of
Doctor Scientiarum Informaticarum

by
Antonio Ridi

from
Italy

Thesis N° 1958
UniPrint, Fribourg
2016

Accepted by the Faculty of Science of the University of Fribourg (Switzerland) upon the recommendation of:

- Prof. Ulrich Ultes-Nitsche, University of Fribourg (Jury President)
- Prof. Rolf Ingold, University of Fribourg (Sponsor)
- Dr. Robert Van Kommer, Innovation Expert at Alliance (Expert)
- Prof. Michael Schumacher, Institute of Business Information System, HES-SO Valais (Expert)
- Prof. Jean Hennebert, Institute of Complex Systems, HES-SO Fribourg (Thesis Director)

Fribourg, March 18^{th}, 2016

Thesis Director	Faculty Dean
Prof. Jean Hennebert	Prof. Fritz Müller

Acknowledgments

This thesis was realized at the HEIA-FR (Haute École d'ingénierie et architecture of Fribourg), in collaboration with the University of Fribourg. I spent an important part of my life, around 4 years, working on this thesis and on several related projects. During this period of time, I had the pleasure to meet several persons who, in their own way, contributed to this thesis.

First and foremost, I am really grateful to my supervisor Prof. Jean Hennebert for the countless things that I learned from him, allowing me to slowly increase, day by day, my scientific background. I really appreciate all the time he spent, especially at the beginning of my thesis, answering my questions, giving me advices and reading my scientific publications. I will always remember our interesting discussions ranging from complex machine learning modeling techniques to the complex, as well, brewing techniques. This constant share of knowledge and of ideas motivated me along all my PhD thesis.

I express my gratitude to Prof. Rolf Ingold, my sponsor at the University of Fribourg, for his precious and interesting advices during our formal and informal meetings. Special thanks also to the experts of this thesis, Prof. Michael Schumacher and Dr. Robert Van Kommer, who spent their time to read, review and provide valuable feedbacks. I would like to mention also professors of the HEIA-FR, in particular Prof. Pascal Bruegger, Prof. Chabbi Drissi Houda, Prof. Pierre Kuonen, Prof. Elena Mugellini and Prof. Omar Abou Khaled for involving me in several activities like courses and projects organized at the HEIA-FR.

I wish to thank Sensimed company, in particular Matteo Leonardi and Thierry Varidel, who gave me the opportunity to collaborate with them on their amazing project. The Chapter on Biomedical data would not have been possible without their inputs. Thanks also to the LESO-PB of the EPFL, in particular the Prof. Nicolas Morel and Nikos Zarkadis, with whom I had the privilege to collaborate and who contributed to the Chapter on the Activity recognition.

Special thanks to my friend and co-worker Christophe with whom I had several enriching discussions and shared numerous scientific papers. A thanks to the colleagues of my first office, Gérôme and Sebastien, and all the others in the open space for making my time at the HEIA-FR so enjoyable. Special mention to Joël and Baptiste for the relaxing pauses and training sessions in our beloved sport.

I am really grateful to Francesco, Leonardo, Maurizio, Simon and Stefano, my friends in the office D30.05, with whom I spent countless enjoyable moments inside and outside the working place. Their friendship is really invaluable to me and I feel lucky to have met all of them.

A thanks to my family in Italy: my parents, Renzo and Mimma, my grandfather, my sisters and their families, in particular my nephews Federico, Dario, Elettra and Maria who made my visit back home always enjoyable and very busy.

Finally a special thank you to my wife Oleksandra, who supported and encouraged me along all my PhD studies and especially during the most difficult moments while writing this thesis.

Version abrégée

Cette thèse aborde l'étude des modèles génératifs pour la classification de séquences temporelles. Le contexte de l'étude est également défini par les signaux acquis dans des applications de monitoring à la maison. Plus précisément, des approches génératives comme les modèles des mélanges gaussiens (GMMs) et les modèles de Markov cachés (HMMs), sont analysés. développés et étendus au-delà de l'état de l'art. Trois sources différentes de données sont considérées: (i) les profils de consommation des appareils électriques pour la reconnaissance de leur type et état, (ii) les données liées à la pression intraoculaire pour la détection de la maladie du glaucome et (iii) les données provenant de capteurs statiques pour la reconnaissance des activités et la prédiction pour la réduction de l'énergie en préservant le confort humain. Plus spécifiquement, la thèse est organisée de la manière suivante:

Chapitre 1: Introduction. Le sujet principal de la thèse est introduit, c'est-à-dire la modélisation générative appliquée aux séquences temporelles des données dans le contexte du monitoring à la maison. Les questions scientifiques ainsi que les aspects principaux de la thèse sont présentés.

Chapitre 2: Fondamentaux. Les concepts fondamentaux de la modlisation générative sont expliqués en détail. La loi de Bayes, la base des approches génératives, est rappelée. Les principes de base des GMMs et HMMs sont également illustrés, en se concentrant sur les algorithmes utilisés pour créer et utiliser ces modèles.

Chapitre 3: Reconnaissance des appareils électriques. Dans ce chapitre, les principes de la reconnaissance des appareils électriques en utilisant leurs profils de consommation sont décrits. Ces profils de consommation sont appelés "signatures électriques". Le design et la collecte d'une base de données, appelée ACS-F et créée dans le cadre de la thèse, sont décrits. Cette base de données est accompagnée de plusieurs protocoles d'analyse, afin qu'il soit possible pour les autres chercheur de comparer leurs résultats. Des approches génératives, comme les GMMs et HMMs, sont utilisées et comparées avec les K-NN, une approche discriminative et non-paramétrique. Les variations du taux de reconnaissance en ajustant les paramètres du système sont systématiquement analysées. De plus, l'impact des coefficients dynamiques est évalué. Quelques applications qui utilisent les propriété naturelles de la modélisation générative sont présentées, par exemple pour l'identification de plusieurs appareils éléctriques additionnés ou la distribution de la complexité computationnelle sur différentes ressources. Finalement, une application supplmentaire des HMMs est proposée. Elle consiste en la récupération de l'information des interactions des utilisateurs avec les appareils éléctriques en utilisant les transitions d'états.

Chapitre 4: Detection du glaucome. Ce chapitre aborde l'identification de la maladie du glaucome en utilisant des informations provenant d'une lentille de contact intelligente (CLS), produite par l'entreprise Sensimed SA. Des caractristiques physiologiques, comme l'amplitude de la pression intraoculaire et la densité des clignements de l'œil, sont calculées à travers un processus d'extraction de caractristiques. Ces caractristiques, combines d'autres, sont utilisées pour la classification des acquisitions du CLS dans deux catégories: patients sains versus glaucomateux. La tâche de détection est realisée en utilisant des modèles de type GMMs et HMMs. Comme dans le chapitre précédent, l'avantage d'ajouter les informations sur la dynamique du signal est évaluée. Les effets de l'ajustement des paramètres des GMMs et des HMMs, comme le seuil de variance, le nombre de gaussiennes et le nombre d'tats pour les HMMs, sont analysés.

Chapitre 5: Reconnaissance des activités pour l'économie d'énergie. Un HMM est utilisé pour la segmentation des données provenant de capteurs de présence. Une spécificité de notre

approche consiste en l'utilisation d'une topologie adéquate pour la reconnaissance des profils des activités. Des modèles plus complexes sont proposés afin de considérer la durée des activités. Plus spécifiquement, des modèles de durées minimales sont utilisés. Ces modelès consistent en la répétition des états des HMMs un certain nombre de fois, en forant l'algorithme à rester un temps minimum dans un état correspondant à une activité. Ces informations sur les activités sont utilisées pour entraîner un ANN qui prdit les activités futures. Finalement, un instrument de simulation estime le gain potentiel d'énergie en contrôlant les apports thermiques en fonction des activités prédites.

Chapitre 6: Conclusions. Finalement, les réponses aux questions scientifiques initialement proposées dans l'introduction sont présentées.

Abstract

In this thesis we describe our work on the use of generative modeling for the classification of time series in the context of in-home monitoring. More precisely, we use generative approaches as Gaussian Mixture models (GMMs) and Hidden Markov models (HMMs).

We use three sources of in-home monitoring data: (i) electrical consumption profiles of appliances for the recognition of their state and category, (ii) intraocular pressure (IOP) related data for the detection of the glaucoma illness and (iii) static sensors data for the activity recognition and prediction aiming at energy consumption savings while preserving human comfort.

More specifically, this thesis is organized as follows:

Chapter 1: Introduction. We introduce the main subject of the thesis, i.e. generative modeling applied to time series, focusing on in-home monitoring data. We propose several scientific questions and present the outline of the thesis.

Chapter 2: Fundamentals. The fundamentals of generative modeling are explained in details. In particular we explain the Bayes law that lies at the bottom of generative approaches. In the following sections we clarify the basic principles of GMMs and HMMs, focusing on the algorithms standing behind.

Chapter 3: Appliance Recognition. In this chapter we describe the principles of the appliance recognition using electricity consumption profiles called "appliance signatures". We report on our work aiming at designing and collecting a database of appliance signatures, called ACS-F, that we made freely available for the scientific community. The database comes along with several analysis protocols, making possible for other researchers to precisely compare their results. We then introduce different generative approaches based on GMMs and HMMs, and we compare them to a non-parametric discriminative approach, the K-NN. We also perform a systematic analysis of the variations in accuracy rate when tuning the parameters of the systems or the features such as adding dynamic coefficients. We provide details about some applications using the natural properties of generative modeling, as the multi-signal identification and the computation distribution among different resources. Finally we provide an additional application of HMMs that consists in retrieving the user interactions by exploiting the state transitions.

Chapter 4: Glaucoma Detection. We analyze a system able to perform the identification of the glaucoma illness using the recordings coming from a wearable contact lens sensor (CLS) produced by Sensimed SA. We firstly focus on the feature extraction in order to retrieve some physiological features, namely the ocular pressure amplitude and the blink density. These and other features are used for classifying the CLS recordings in two categories, healthy and glaucomatous. We use GMMs and HMMs as machine learning techniques. As for the previous chapter, we observe the benefit of adding information about the evolution of the signal. We also analyze and interpret the effect of tuning some of the GMMs/HMMs parameters, as the variance floor, the number of Gaussians and the number of states for the HMMs.

Chapter 5: Activity Recognition for Energy Savings. In this chapter we use HMMs for segmenting data coming from presence detection sensors. The aim is to perform Activity recognition, using a state topology adequate to activity patterns. We compare the performances of standard HMMs against a more complex modeling technique that considers the duration of the activities. We opt for a strategy of minimum duration modeling that consists in repeating a certain number of times the states of HMMs, forcing the algorithm to spend a minimum period

of time in a given state. We use the information of the activities for feeding an ANN that, in turn, predicts future activities. Finally we use a building physics simulation tool for estimating the potential energy savings when predicting future activities.

Chapter 6: Conclusions. We draw general conclusions and we summarize the answers to the scientific questions proposed in the Introduction.

Contents

Symbols and Acronyms

We present the symbols and the acronyms used throughout this thesis.

List and description of symbols

Machine learning	
$\mathcal{O}_{tr}$	The set of observations in the training set
$\mathcal{O}_{ts}$	The set of observations in the test set
$\mathcal{D}_{tr}$	The set of feature vectors in the training set
$\mathcal{D}_{ts}$	The set of feature vectors in the test set
$\mathcal{C}$	The set of classes
U	Total number of observations in the training set
V	Total number of observations in the test set
M	Total number of classes
D	Dimension of the feature space

Estimation of probabilities	
θ	Vector of parameters
l	Log-likelihood function
P	Dimension of the feature vector θ
$\mathcal{Z}$	The set of latent variables
K	The number of clusters and the number of mixtures
J	The within-cluster sum of squares in K-Means
$\boldsymbol{\mu}$	The mean vector
$\boldsymbol{\Sigma}$	The covariance matrix
$\boldsymbol{w}$	The vector of weights in the Mixture models
$\boldsymbol{\theta}$	The set of parameter vectors
ζ_{um}	Observations-class relation
ζ_{uk}	Observations-cluster relation
r_{uk}	Responsibility observations-cluster

State-based notation	
$\mathbf{O}$	The set of observations as time series
$\mathbf{X}$	The feature vectors in a time series
$\mathbf{x}_t$	The feature vector at a given discrete time t
T	The length of a time series
τ	The index set
$\mathcal{S}$	The set of states
N	Total number of states
$\mathcal{Q}$	State sequence
λ	Hidden Markov model
A	Transition matrix
A_i	Set of out-going transitions of the S_i state
B	Vector of function containing the emission probabilities
π	Initial state probabilities
$\boldsymbol{\eta}$	The vector of symbols for a discrete HMM

H	Total number of symbols for a discrete model
$\boldsymbol{\alpha}$	Vector of the forward probabilities
$\boldsymbol{\beta}$	Vector of the backward probabilities
δ	The vector of highest probabilities along a single path
ψ	The vector of paths with the highest probabilities
γ	Probability of being at the state S_i at time t
ξ	Probabilities of being in the state S_i at the time t and at the state S_j at the time $t+1$
ν	Variation floor

Symbols	
$\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_F\}$	List of F elements in the set $\mathcal{E}$
$\mathbf{e}_f$	Specific element in the set $\mathcal{E}$
$\mathbf{e}_t$	Specific element in the time series $\mathbf{E}$
$\hat{e}$	Estimation of the element e
E'	Transposition of the matrix E
∇_e	Gradient operator

Special characters	
$\mathcal{N}$	Gaussian function
$\mathbb{R}$	Real space
p	Probability
O	Complexity
ϵ	Small value for testing the convergence of an algorithm

Chapter 1

Introduction

Contents

1.1 Global context: generative modeling applied to time series

Big data has been recently emerging as a popular trend, which consists in managing and processing large quantity of data. Referred to as the "black gold" of the new century, Big data represents an opportunity for companies and organizations that are capable of mining this data. Further to the issues related to the practical management of large quantities of data, an important branch of Big data is the elaboration of mathematical models of knowledge in a data driven way. This is called machine learning and already raised in popularity before the mainstream development of what is called today Big data.

Machine learning can be thought as the science of making the computers able to fulfill desired tasks without explicitly programming them. Nowadays, machine learning becomes increasingly pervasive and people are using machine learning applications several times a day without even realizing it. Machine learning applications range from the detection of intruders to self-driving cars, from the detection of spam to the intelligent personal assistant on our smart-phones. Generally speaking, several problems are addressed by machine learning, as classification, clustering, regression or anomaly detection.

Machine learning applications commonly deal with temporal data, that is typically generated by a process evolving with time. Temporal data is usually generated by an hardware system, as an environmental sensor, or software system, as a financial platform. Temporal data is discrete or continuous. In the first case, the process generates data depending on external events, as for instance an environmental sensor recording the time-stamps of motion movements. Temporal data in this case is called *data sequence*. In the second case, the process generates data by using a fixed time step, as for instance a sensor that recurrently measures the temperature every five minutes. Temporal data in this case is called *time series*. A time series can also be thought of as a sequence of observations equally spaced in time. In this thesis we work with time series, eventually transforming *data sequences* in *time series* before being processed.

Time series have several applications, e.g. to predict the future based on the past, to understand the underlying process or to classify unlabeled time series to some predefined classes. In this thesis we focus on classification tasks applied to time series. Time series classifications can be defined in two different ways depending if we are interested in classifying each observation in the time series or the whole time series. In the first case a label is associated to each observation, while in the second

case a label is associated to the whole time series. The whole time series classification could also be obtained from the single observation classification, for instance by using a majority voting algorithm on the labels associated to the observations. In this thesis we focus on the classification of the whole time series.

In machine learning two main approaches are used for classification tasks: generative or discriminative approaches. In the first approach, the model is able to generate new observations respecting the underlying distributions. The principle is, given multiple classes, to try to find which category most likely has generated an observation. In the second approach, the aim is to discriminate between classes without attempting to model the underlying probability distributions. In practice, it is not important how observations are generated, it only matters how categorize the new observations. The two approaches have many different implications. Usually the discriminative approach is know to provide better performances, given that the classification problem is directly solved without previously modeling the underlying distributions. On the other hand, generative modeling offers some interesting properties. Firstly, it provides a better robustness to the scalability, i.e. when increasing the number of classes. Secondly, it provides a better robustness to the adaptability, i.e. when changing the prior distribution of the classes. Finally, it has a natural "distributability", i.e. when distributing the computation among different resources. In addition, when using generative modeling as Gaussian Mixture models (GMMs) and Hidden Markov models (HMMs), time series are intrinsically adequate.

This thesis has been elaborated along two projects: (i) Green-Mod and (ii) Sensimed Data Intelligence. The first project aimed at the elaboration of Information Systems able to optimize the energy consumption in buildings while preserving human comfort. In particular, the project focused on the use of state-based stochastic modeling applied to temporal signals acquired from heterogeneous sources such as distributed sensors and user triggered events. The project was realized in collaboration with the Laboratoire d'Energie Solaire et de Physique du Bâtiment (LESO-PB) at the Ecole Polytechnique Fédérale de Lausanne (EPFL). The Chapter 3 and 5, that will be later presented, are partially related to this project. The second project focused on the glaucoma detection by processing data coming from a smart contact lens produced by Sensimed SA. The contact lens measures during 24-hours the circumferential deformations of the eyeball. This project focused on the use of machine learning for the identification of different glaucoma varieties, for the prediction of glaucoma development and for a proactive treatment of the glaucoma through a reliable feature extraction. The Chapter 4 is related to this project.

According to the content of these projects, the backbone that we developped in this thesis is the analysis and development of novel generative modeling approaches for data recorded as time series. A focus is given on GMMs and HMMs approaches, that are known to provide good performances on time series. In the next section, we provide more information about in-home monitoring applications which are the specific context of this thesis.

1.2 Specific context: in-home monitoring

As a matter of facts, people spend a large part of their lives inside buildings, in particular their homes. A big quantity of data can be continuously captured from buildings and from people living inside their homes. The information extracted from this data can be used in the advantage of the inhabitants, for instance improving their comfort, their security or optimizing the energy consumption.

In Figure 1.1 we illustrate different sources of data inside a smart house:

- Static sensors are becoming more and more popular because of their reduction of size and weight and also thanks to the reduction of their cost. Nowadays, static sensors as motion detectors, can be commonly found in private and public buildings.
- Wearable devices are also gaining popularity, as smart-phones, smart watches and other similar devices. We observe that in the last decades, people are getting used to constantly wear these devices.
- Biomedical sensors are more and more used for monitoring the health condition of a patients at home. The popularity of in-home health monitoring has increased, mainly caused by the need of reduction of the hospitalization costs and the improvement of the quality of life of patients. Wearable devices able to locally record patient data are largely used.

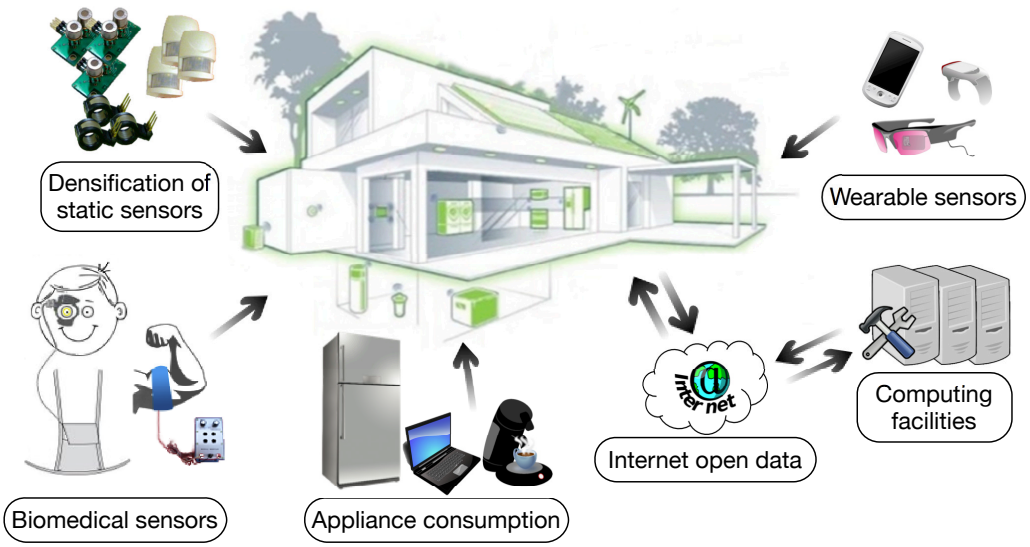


Figure 1.1: A smart house with different sources of data, as static sensors, wearable devices, biomedical sensors and appliance consumption profiles coming from energy meters. Computing facilities could help to manage large quantity of data.

- Smart energy meters are able to measure the whole house consumption or that of single appliances. Such information is particularly useful for understanding the energy sources of consumption and then applying energy saving measures where possible.

In-home data has to be stored somewhere before being processed. It is usually stored remotely or locally. In the first case, data is easily accessible, without the need to be physically in place. Computing facilities, as clusters for data analysis, could be used for managing large quantity of data.

This thesis analyses three applications in the context of in-home monitoring. We pragmatically choose as data sources signals at our disposal in the projects that allowed the realization of this thesis. Data consists in time series on which we systematically perform a classification task. We use three sources of in-home monitoring data: (i) electric consumption signatures of appliances for identifying the state and category, (ii) intraocular pressure (IOP) related data for the detection of the glaucoma illness and (iii) static sensors data for the activity recognition and the prediction of future activities. In the first case, electrical signatures are supposed to work well with a generative approach as HMMs given the state-based nature of appliances. Given the high number of appliance categories and the continuous evolution of the market, it is very likely that new appliance categories are added and removed over time. A system working with such data could benefit from generative modeling that avoids a fully re-training when a new appliance category is added of removed. Moreover, some of the electrical features have an additive property. As will be shown, this property can be favorably leveraged with generative modeling. In the second case, the glaucoma illness is expressed through different categories. As in the previous case, it is useful to be able to model new classes of glaucoma without fully re-training all the system. Moreover, the healthcare professionals should be able to interpret all machine learning steps for working with data scientists. When using generative modeling, the score associated to each IOP related sequence has a mathematical and interpretable meaning that can be easily used even from a person not working in the data science field. This consideration is certainly valid for other biomedical signals. In the third case, static sensor data works well with a generative approach as HMMs as already shown in literature, where they are used for the Activity recognition. However, one of the limitations of HMMs is that the duration of the activity are not considered. Such system could benefit of the inclusion of the duration.

1.3 Constitutive hypothesis and research questions

In this thesis we use generative modeling based on stochastic approaches. Using such stochastic approaches, we are able to estimate the probability distributions of the input observations. For instance, GMMs are able to estimate the probability distributions as a weighted sum of Gaussian functions. The times series under analysis are likely to have a temporal and state-based nature. A natural extension to GMMs when attempting to capture a notion of states is HMMs. HMMs are able to automatically detect the natural states of the signals and they should theoretically provide a better parameterization.

Stochastic modeling, as GMMs and HMMs, has been shown to provide good results in different fields, as in speech, handwriting and gesture recognition. Our constitutive hypothesis is that such modeling techniques also apply to other types of signals acquired through in-home monitoring systems. Comparing the performances of generative methods against discriminative approaches has already been widely done in literature. In this thesis, we therefore decided to focus on the advantages of using generative modeling.

Generally speaking, HMMs can be considered as a generalization of GMMs where the hidden variables are related through a Markov process. The Markov process is a stochastic model having the Markov property: the process is "memoryless", i.e. the future states of the process depend only on the present state and not on the previous sequence of states. Another important assumption implicitly made when using HMMs is that the current observation is independent of the past and future observations. While the Markovian hypothesis is usually fair enough, often the assumption about the observation independence represents an important weakness of HMMs.

Under the assumption that GMMs/HMMs approaches provide favorable results for the classification of in-home monitoring time series, we propose scientific questions that will be addressed in the thesis:

1. What are the benefits in using generative approaches, such as GMMs and HMMs, in the context of time series acquired in in-home monitoring applications?
2. Can we use the natural properties of generative approaches for performing tasks otherwise impossible or hard to make when using discriminative approaches?
3. Is the notion of states, as captured by HMMs, beneficial to the classification task, especially when compared to stateless approaches, such as GMMs?
4. Do we have way to relax consistently the observation independence assumption of GMMs/HMMs systems?
5. Can we somehow relax the Markovian hypothesis of HMMs systems by including the information about the duration of the observation sequences?

Further to this set of more fundamental questions, we also address specific questions related to the type of signal under analysis:

1. A basic and preliminary question but not explored so far in the appliance recognition context: is it possible to analyze the appliance signatures allowing the appliance identification?
2. Can we discriminate healthy and glaucomatous patients? Do we have benefits in using states related to the patients' activities, such as sleep and active periods, for the detection of glaucoma?
3. Can we benefit from duration models, that can be easily integrated in generative models, to predict with a level of performance that allows saving energy?

1.4 Outline of the report

- **Chapter 1: Introduction**.

  This chapter is divided in four parts. Firstly we introduce the general context, i.e. the generative model applied to time series. Then we focus on the specific context, i.e. the in-home monitoring,

and we introduce the underlying background of the thesis. Then we present the constitutive hypothesis and the research questions besides this thesis. Finally in this section we present the outline of the report.

- **Chapter 2: Fundamentals**.

  Chapter 2 starts with a description of the standard machine learning workflow, giving an insight on the bases of state-based modeling. Then we introduce the Bayes' law, that is very helpful for describing the main differences between discriminative and generative approaches. We explain in details the terms of the Bayes' law, i.e. the *posteriors*, the *priors* and the *evidence*, and how these terms are estimated, focusing on multivariate Normal distributions. Later we introduce the Mixture model, by providing details about the most used algorithm for its initialization, i.e. the K-Means, and the main algorithm standing beside its training, i.e. the Expectation-Maximization algorithm. In particular we focus on GMMs, by far the most used Mixture model. Following the natural flow of the discourse, we introduce HMMs. In particular we show how to model the underlying stochastic processes for computing the transition and emission probabilities. We deepen into the three canonical problems of HMMs, namely the evaluation problem, the decoding problem and the learning problem, and we provide a list of the main hypotheses standing behind HMMs. We conclude the chapter by showing the similarities between GMMs and HMMs.

- **Chapter 3: Appliance recognition**.

  In this chapter we describe the appliance recognition process using the appliance energy consumptions, also called "appliance signatures". We firstly introduce the subject and the state of the art. In particular we present the law of the electricity and we compare the data acquisition, feature extraction and machine learning techniques previously analyzed in literature. Later we present our database, containing several appliance signatures that we have recorded and used for the appliance recognition task. The data acquisition protocol and the data formats are also presented. In order to be able to compare our results with the ones of other researchers, we propose two categories of evaluation protocols, called "fixed time length" and "variable time length". The first allows to use the whole duration of the signatures, namely 1 hour, while the second analyzes shorter portions of the signatures. Two specific protocols, called "intersession protocol" and "unseen appliance protocol", are proposed within the categories of evaluation protocols. The first classifies appliances already seen in the training set, while the opposite occurs when using the other protocol. Later we present the results of the classification task when using GMMs and HMMs. In order to compare the results, we apply a third classification algorithm, namely the K-NN. For each algorithm we evaluate the influence of the dynamic coefficients. Then we present the advantages of using a generative approach, for instance by reducing the number of classes, performing a multi-signal classification and distributing the computation among several devices. We conclude the chapter by presenting our work on the detection of user activity interactions by analyzing the transitions between states of the appliances when using HMMs.

- **Chapter 4: Glaucoma Detection**.

  Chapter 4 presents the analysis of physiological data of patients recorded in-home, i.e. outside the conventional clinical settings. We start by introducing the subject and the state of the art. We discuss about several in-home health monitoring applications and we analyze the state-based modeling techniques used in literature on physiological signals. In particular, we focus on works dealing with signals captured by the human eye, as the continuous monitoring of the intraocular pressure (IOP). Later we explain the specific classification task tackled in this chapter, i.e. the identification of the glaucoma illness by using a wearable contact lens sensor (CLS) produced by Sensimed SA. The device, called Triggerfish®, is capable of recording up to 24-hours of circumferential changes in the area of the corneoscleral junction of the eye. Actually, the only risk factors for the glaucoma largely accepted by the ophthalmologic community are the age, the family history and the intraocular pressure level over a specific threshold (21 mmHg). A strong correlation between the intraocular pressure and circumferential changes has been demonstrated by Sensimed SA. We present in details our workflow for the glaucoma detection,

including the evaluation protocol. We firstly focus on the feature extraction allowing to achieve two physiological features, namely the ocular pulse amplitude and the blink density. Later we apply the generative approach GMMs/HMMs and we tune some of their parameters, as the variance floor or the number of Gaussians. Following the same procedure of the previous chapter, we evaluate the influence of the dynamic coefficients. Finally we perform the feature selection on the data for avoiding the redundant and non-informative features.

- **Chapter 5: Activity Recognition for Energy Savings**.

  In this chapter we describe the activity recognition and prediction tasks aiming to control building equipments for energy saving. As for the previous chapters, we firstly introduce the subject and the state of the art. In particular we present the definition of Smart Homes and the works in literature using state-based modeling in Smart Homes. Later we describe our workflow, mainly divided in three steps: the activity recognition, the activity prediction and the energy saving simulation. The activity recognition is performed by using two approaches: standard HMMs and minimum duration modeling. The minimum duration modeling consists in modifying the topology of the HMMs after a standard training for capturing the information about the activity duration. We use three different solutions in order to limit the quantity of data in the training set: the *random selection training*, the *specific shift training* and the *complete shift training*. For the activity prediction we apply an ANN on the labels produced during the activity recognition step. We compare this approach with two other methods using a HMM and an ANN directly on features. Then we apply energy saving potential measures by using the information about the predicted activities and we present the results in terms of energy saving.

- **Chapter 6: Conclusions**.

  In Chapter 6 we conclude the thesis. In the Conclusions we recap the scientific questions proposed in this chapter, providing detailed answers.

Chapter 2

Fundamentals

Contents

2.1 Introduction to Machine Learning

The core of this thesis is the application of Machine Learning algorithms on in-home monitoring data. Machine Learning has been defined as the science of getting computers to act without being explicitly programmed [1]. Another definition could be the computation of mathematical models from exemplary data to solve complex tasks on new, unseen, data.

Other terms are currently related to "Machine Learning", as Pattern Recognition, Data Mining and Knowledge Discovery in Databases, even if some subtle differences exist between these terms. Pattern Recognition focuses on the identification of pattern or regularities in data, while Data Mining and Knowledge Discovery in Databases usually involve the data management aspect of databases.

Two tasks are typically realized by Machine Learning: *classification* and *regression*. The *classification* associates the input data with one among multiple categories, while the *regression* computes a continuous variable as output. Depending on the availability of the expected outputs (also called *ground truth*), there are three different approaches [2]:

- *Supervised learning.* Input data and expected outputs are given to the machine that is able to infer the relation between inputs and outputs. Depending on the type of output, we distinguish two categories: (i) *classification* if the output is a finite label, (ii) *regression* if the output is a continuous variable. A variant of the classification problem consists in relating each input data to multiple classes, called *multi-label classification*.

- *Unsupervised learning.* Only input data is given to the machine and, without using any additional information, the underlying data structure has to be found. We distinguish two approaches: (i) *clustering*, aiming at grouping input data with similar characteristics and (ii) *latent variable modeling* with the goal of relating input data with a set of latent variables.

- *Reinforcement learning.* Considering a dynamic environment where a specific goal has to be attained, i.e. safely driving a car, the machine has to attain the result without any information about the proximity to the goal.

In this thesis we focus on classification problems in supervised leaning. As previously said, in *supervised learning* we have pairs of input data and expected outputs. We want the machine to be able to learn the existing relations between input data and expected outputs. Inputs usually come from a system under investigation and are usually called "observations". Given that we face the classification problem, the expected outputs are categories, also called "classes" or "labels".

The first step consists in separating data in two sets: *training set* and *test set*. Observations and labels contained in the training set are used by machine learning algorithms for learning their relation. This procedure is also known as "computing a model". The test set is used for evaluating the capacity of the machine learning algorithm to classify observations never seen before by classifiers. The subdivision between training and test set depends on several factors, e.g. the original size of data or the type of machine learning algorithm used. Limiting the subdivision into two sets could be a problem, especially for machine learning algorithms having several parameters that can be tuned. In fact, we could train the classifiers by systematically modifying the parameters for obtaining the best results on the test set. In this case we could experience the problem of *overfitting*, namely when we create an overly complex system that is unlikely to perform well on new patterns [2]. To avoid this problem, a cross-validation procedure is commonly used [3]. The goal of the cross-validation is to test the model during the training phase by using an independent set, usually called *validation set*. The result of the classification is influenced by the random partitioning into subsets. To reduce the variability of the result, the classification task can be repeated several times by using different partitions and averaging the results over the rounds.

When data and labels have been separated, we can proceed with the *training phase*, as illustrated in Figure 2.1. The whole set of observations $\mathcal{O}$ is split into a training set $\mathcal{O}_{tr} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_U\}$ and a test set $\mathcal{O}_{ts} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_V\}$, where U and V are respectively the total number of observations

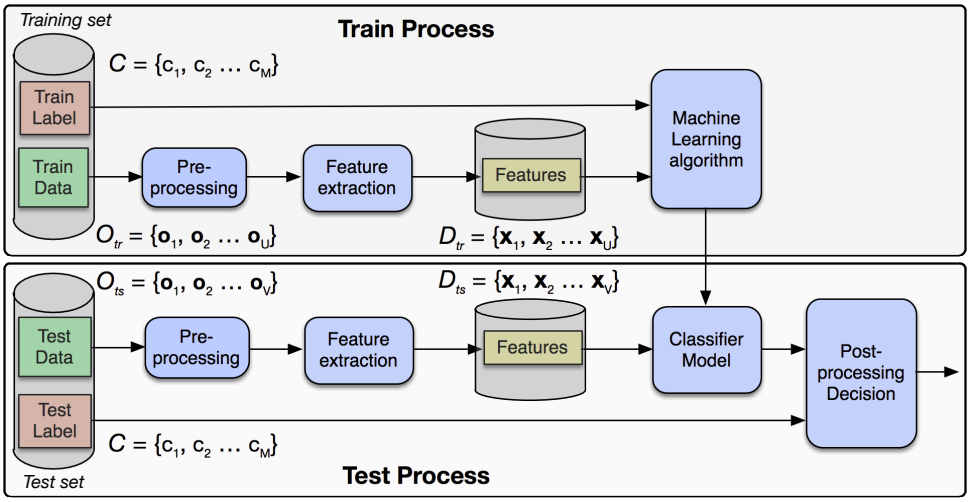
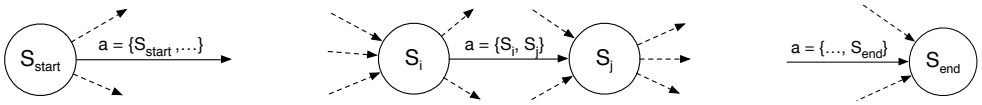


Figure 2.1: The basic schema of supervised machine learning for the classification task. The whole set of observations $\mathcal{O}$ is split into a training set $\mathcal{O}_{tr}$ and a test set $\mathcal{O}_{ts}$. Both sets are pre-processed and the observations are transformed into *feature vectors*, respectively $\mathcal{D}_{tr}$ and $\mathcal{D}_{ts}$. A model is usually generated by using $\mathcal{D}_{tr}$ and the list of labels. The model is evaluated in the *testing phase*.



(a) Start state (b) Generic states (c) End state

Figure 2.2: Three types of state: (a) start state, (b) generic states and (c) end state.

in the training and test set. We indicate the set of possible classes with $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$, where M is the total number of classes. The first step consists in *pre-processing* the raw observations. Typically the observations are "cleaned", e.g. through different normalization, filtering or removal procedures of outliers. The "cleaned" observations are then sent to the *feature extraction* block. The aim is to find a better representation of data for simplifying the work of the machine learning algorithm. Typically redundant and useless information is filtered and information not explicitly available is computed. The feature extraction is considerably different from the pre-processing given that it changes the data representation. The observations are transformed into *feature vectors*, that we indicate as $\mathcal{D}_{tr} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_U\}$. Every feature vector belongs to a D-dimensional space $\mathbb{R}^D$, called *feature space*. The set of feature vectors $\mathcal{D}_{tr}$ is sent to the machine learning algorithm with the list of labels. The aim of the classifier is to build a model describing the relation between feature vectors and labels. The model here generated is evaluated in the *testing phase*.

Suppose to have an observation $\mathbf{o}_v$ belonging to the *test set* $\mathcal{O}_{ts} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_V\}$ where V is the total number of observations in the test set. By applying the same pre-processing and feature extraction operations, we obtain a feature vector $\mathbf{x}_v \in \mathcal{D}_{ts} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_V\}$. In a classification problem the objective is to determine to which class c_m the new feature vector belongs. In practice, we want to compute $p(c_m \mid \mathbf{x}_v)$ for each class c_m, and then assign $\mathbf{x}_v$ to one of the classes. Using the Bayes' law, the decision is taken by comparing the probabilities and selecting the most probable one.

2.1.1 Overview of state-based modeling approaches

State-based modeling can be thought of as a way of representing a system under investigation by using a set of states. How these states are defined, which are the typical parameters and the relations between the states, constitute the mathematical representation of state-based modeling. Usually dynamic systems are represented by one state for a given time and a change of the system usually entails a transition to another state in the model representation. State-based modeling is transversely used in several domains, e.g. electronic design automation [4], communication protocol design [5] and language parsing [6]. Moreover state-based modeling can be used to mathematically represent time-series data produced by systems with multiple modes. The signal is said to be piece-wise stationary, in the sense that in a given state the data emission parameters are supposed to be somehow stables. State-based modeling is generally useful in all the domains that typically deal with sequences of data points emitted by natural systems in time such as speech [7], handwriting signatures [8] and financial time-series [9]. In this thesis, state-based models are applied to signals produced by in-home monitoring systems.

Nonetheless, what are exactly *state-based models* (SBM) ? Several synonyms or slightly different terms are used depending on the context: state-based machine, extended finite-state machine, etc. [10]. We can hereunder report a well-known definition : SBM *consists of a finite set of states (a non-strict subset of which are start states), a set of events (or inputs) and a transition function that, based on the current state and event, determines the next state (i.e. performs transitions between states)* [10].

The most common approach for describing a SBM is using graphs. In graph theory two main classes exist: (i) undirected graphs and (ii) directed graphs. Undirected graphs consist in a set of nodes (or vertex) and a set of edges connecting the nodes. Directed graphs (or digraph) consist in a set of nodes and a set of arrows (or arcs). An arrow can be represented as a pair $a = \{x, y\}$, indicating that the arrow is directed **from** x (the *tail*) **to** y (the *head*). Using such representation, we distinguish three types of states, as illustrated in Figure 2.2:

- *Start states* are representing entry points in the model. The arrows are directed out of the node and have all the *tails* equal to S_{start}.

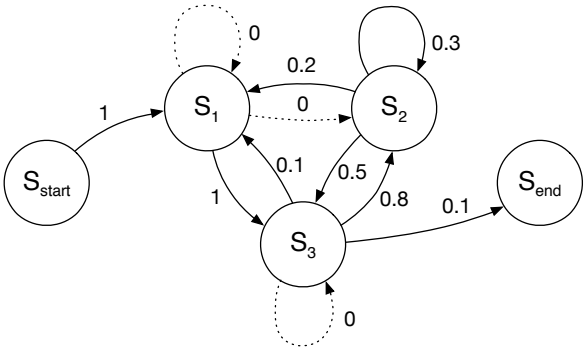


Figure 2.3: Example of a probabilistic SBM, where all the transitions probabilities are between 0 and 1 and for each state the sum of the outgoing transitions probabilities is equal to 1. The arrows with probability equal to 0 have been dotted. The dotted lines from the start and to the end state have not been drawn.

- *Generic states* are all the other nodes where arrows can enter and exit. Such states have at least one arrow as input and at least one arrow as output.
- *End states* represent exit points in the model. The arrows are directed to the node and have all the *heads* equal to S_{end}.

One of the most common type of graph is the weighted graph. In that case, the edges of the graph are associated with a weight, that usually is a real number. In some cases certain restrictions on the weights are applied. For instance, if the weights are probabilities, they have to be real numbers between 0 and 1. SBM can indeed be described with a *probabilistic* point of view to describe a *stochastic* (or *random*) process. Given a state S_i having N transitions to N different states (outgoing transitions), each transition is associated to a probability. As a consequence, for a given input a path can be taken with a given probability. The set of transitions $A_i = \{a_{i1}, a_{i2}, \ldots, a_{iN}\}$ of the S_i state has the following properties:

- All the transitions in the set have a probability between 0 and 1:

$$\forall a_{ij} \in A_i \mid 1 \leq i, j \leq N \implies 0 \leq a_{ij} \leq 1 \tag{2.1}$$

- The sum of all the outgoing transitions is equal to 1:

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.2}$$

In Figure 2.3 an example of *probabilistic* SBM is given.

With a *probabilistic* SBM, for a given input we are not forced to follow a specific path and neither to choose one among all the possible paths. All paths have a given probability to be taken. The most common *probabilistic* SBM is the Hidden Markov model (HMM), that will be analyzed in details in Section 2.5. Mixture models can be considered as degenerate case of HMM having only one state. We will focus in this thesis on Gaussian Mixture models (GMMs) that will be analyzed in Section 2.4.

2.2 Bayes' law

The classification problem typically can be split in two steps when using a supervised learning approach: *inference* and *decision*. During the *inference* process, training data is used to determine the joint distribution $p(\mathbf{x}, c_m)$ that completely describes the relation between the input vector and the classes in terms of probabilities. In practice, the probabilities of the input vector $\mathbf{x}$ to belong to a

class c_m are computed. During the *decision* process the choice of the best class for the input vector is taken.

By using the conditional probability formula and the symmetric property of the joint probabilities[1]:

$$\begin{cases} p(\mathbf{x}, c_m) = p(\mathbf{x} \mid c_m)p(c_m) \\ p(\mathbf{x}, c_m) = p(c_m, \mathbf{x}) \end{cases} \tag{2.3}$$

we obtain the Bayes' law:

$$p(c_m \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c_m)p(c_m)}{p(\mathbf{x})} \tag{2.4}$$

where $p(c_m \mid \mathbf{x})$ is the *a posteriori* (or *posterior*) probabilities, $p(c_m)$ is the *a priori* probability (or *prior* probability) of the class c_m, $p(\mathbf{x} \mid c_m)$ is the likelihood of c_m with respect of $\mathbf{x}$ and $p(\mathbf{x})$ is the evidence factor (or *evidence*). The posterior probability indicates the probability of c_m being the matching class given the feature vector $\mathbf{x}$. The prior probability indicates how likely the feature vector $\mathbf{x}$ belongs to the class c_m independently of $\mathbf{x}$. For instance, if the classes are strongly unbalanced, the most represented classes will have a higher a priori probability. The likelihood indicates how much the feature vector $\mathbf{x}$ is likely in the population of the class c_m. The evidence factor scales the posterior probability to sum to one. When comparing the posterior probabilities for choosing the most probable class, the evidence factor can be overlooked given its independence from the class type. The Formula 2.4 shows how the *a posteriori* probability, *a priori* probability, the likelihood and the evidence factor are related.

Another representation of the Bayes' law is obtained by rewriting the evidence:

$$p(c_m \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c_m)p(c_m)}{\sum_m p(\mathbf{x} \mid c_m)p(c_m)} \tag{2.5}$$

Generally speaking, the classification problem can be solved by modeling (i) the likelihoods and prior probabilities or (ii) directly the posterior probabilities. The first approach is called *generative* modeling and it consists in explicitly or implicitly modeling the probability distributions of the observed data. Machine learning algorithms such as Naive Bayes classifiers, Gaussian Mixture models and Hidden Markov models belong to this category. The term "generative" indicates that the model is also able to generate new observations respecting the underlying distributions. The second approach is called *discriminative* (or conditional) modeling. Machine learning algorithms such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs) and Multinomial logistic regression belong to this category. The term "discriminative" indicates that the aim is to discriminate between classes without attempting to model the underlying probability distributions.

In practice discriminative modeling directly estimates the posterior probabilities through the computation of functions associated to class-boundaries. It could be said that discriminative modeling directly solves a problem (the classification) without before solving a more general problem (finding the underlying probability distributions). For this reason, the discriminative approach usually yields better performances [11]. On the other hand, generative modeling is more flexible and it is reported to better deal with missing input values, outliers and unlabeled data [12]. In addition, generative models do not need to be fully retrained when there is a new class or label, while this is needed for discriminative models.

In the following sections we present and explain the Maximum-likelihood estimation (Section 2.3), Mixture models (Section 2.4) and Hidden Markov models (Section 2.5). The next sections have been inspired by the books *Pattern Classification* of *Duda et al.* [2], *Pattern Recognition and Machine Learning* of *Bishop* [3] and *Machine Learning: A Probabilistic Perspective* of *Murphy* [13]. In addition, the Section 2.5 has been inspired by the article *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition* of *Rabiner* [14] and the PhD thesis *Hidden Markov Models and Artificial Neural Networks for Speech and Speaker Recognition* of *Hennebert* [15].

[1] In this thesis we use the lowercase $p(\cdot)$ for indicating both the probability mass functions and the probability density functions

2.3 Maximum-likelihood estimation

An optimal classifier could be built if one could compute the true *prior* probabilities $p(c_m)$ and the class-conditional densities $p(x \mid c_m)$. This information is usually not available for natural phenomena that we observe in real-life. However, for supervised pattern classification problems, the probabilities and densities of probabilities are estimated by using data pooled in a training set.

The estimation of prior probabilities is a rather simple task. Considering U observations in the training set, we can define ζ_{um} equal to 1 if the ground truth for the u-th sample is c_m and equal to 0 otherwise. The prior probability can then be written as:

$$\hat{p}(c_m) = \frac{1}{U} \sum_{u=1}^{U} \zeta_{um} \tag{2.6}$$

In other terms, the priors are proportional to their representativeness in the training set.

Estimating the probability density functions is a more difficult task. Usually this problem is simplified by making some hypotheses on the form of the probability distribution. In the case of a Normal distribution, $p(\mathbf{x} \mid c_m)$ is estimated through the computation of the parameters of the density distributions, i.e. the mean and covariance matrix. As for the estimation of the prior probabilities, data in the training set are used for estimating the density parameters.

In practice, we have to split the training set $\mathcal{D}_{tr}$ into different sets depending on their class and we obtain a set of collections of samples $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$. Assuming that $p(\mathbf{x} \mid c_m)$ has a parametric form, it can be represented by a parameter vector θ_m. The dependence of the likelihood on the parameter vector θ_m can be made explicit by including the latter term in the expression of the likelihood: $p(\mathbf{x} \mid c_m) = p(\mathbf{x} \mid c_m, \theta_m)$. The problem of estimating the density of probabilities now becomes the problem of estimating the values of the parameter vectors $\{\theta_1, \theta_2, \ldots, \theta_M\}$. For instance, when considering a Normal distribution, then $\theta_m = \{\mu_m, \Sigma_m\}$, where μ_m is the mean vector and Σ_m the covariance matrix of the m-th class. We can reasonably assume that the estimation of the vector of parameters θ_m of the m-th class is independent of the samples belonging to the other classes. As a consequence, for estimating θ_m only the U_m samples belonging to the relative collection $\mathcal{D}_m$ are necessary. For sake of simplicity the subscript m will be omitted from $\mathcal{D}_m$, θ_m and U_m in the further formulas.

Suppose that the collection $\mathcal{D}$ contains U samples $\{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_U\}$. By assuming independence between the samples, we can compute the likelihood as:

$$p(\mathcal{D} \mid \theta) = \prod_{u=1}^{U} p(\mathbf{x}_u \mid \theta) \tag{2.7}$$

The maximum-likelihood estimate of θ is then the value $\hat{\theta}$ that maximizes $p(\mathcal{D} \mid \theta)$. Even if the independence assumption is often incorrect, it is needed to make the problem tractable [16]. The value of the probability can attain a very low value, therefore working with the logarithm of the probabilities is more practical. Given that the logarithm is monotonically increasing, we are guaranteed that the $\hat{\theta}$ that maximizes $p(\mathcal{D} \mid \theta)$ also maximizes $\ln p(\mathcal{D} \mid \theta)$. We can define a log-likelihood function $l(\theta)$:

$$l(\theta) \equiv \ln p(\mathcal{D} \mid \theta) \tag{2.8}$$

The maximum-likelihood estimate can be written as:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ l(\theta) \tag{2.9}$$

The Formula 2.7 can be rewritten by using the logarithm:

$$l(\theta) = \sum_{u=1}^{U} \ln p(\mathbf{x}_u \mid \theta) \tag{2.10}$$

Assuming P the number of parameters in the vector θ, we can define the gradient operator ∇_θ as:

$$\nabla_\theta = \begin{bmatrix} \frac{\partial}{\partial \theta^1} \\ \vdots \\ \frac{\partial}{\partial \theta^P} \end{bmatrix} \tag{2.11}$$

We can apply the gradient operator ∇_θ to the Formula 2.10:

$$\nabla_\theta l = \sum_{u=1}^{U} \nabla_\theta \ln p(\mathbf{x}_u \mid \theta) \tag{2.12}$$

For the maximum-likelihood estimation we can set the following condition:

$$\nabla_\theta l = 0 \tag{2.13}$$

The solution of this equation, $\hat{\theta}$, can be a local maximum, a local minimum or an inflection point of $l(\theta)$. However we should consider that, due to the limited amount of training samples, $\hat{\theta}$ is an estimation of the probability density function. In the next section we report the example of the estimation of the density parameters for a multivariate Normal distribution.

2.3.1 Multivariate Normal distribution

In the multivariate Normal case, the parameter vector θ is equal to $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix. Hereunder we report the formula of the multivariate Normal distribution:

$$p(\mathbf{x}_u \mid \theta) = p(\mathbf{x}_u \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} exp[-\frac{1}{2}(\mathbf{x}_u - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x}_u - \boldsymbol{\mu})] \tag{2.14}$$

The logarithm of the multivariate normal distribution is:

$$\ln p(\mathbf{x}_u \mid \theta) = -\frac{1}{2} \ln[(2\pi)^D |\boldsymbol{\Sigma}|] - \frac{1}{2}(\mathbf{x}_u - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x}_u - \boldsymbol{\mu}) \tag{2.15}$$

We can apply the gradient operator ∇_θ to the previous formula using the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively:

$$\nabla_\theta \ln p(\mathbf{x}_u \mid \theta) = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_u - \boldsymbol{\mu}) \\ -\frac{1}{2}[\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}(\mathbf{x}_u - \boldsymbol{\mu})(\mathbf{x}_u - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1}] \end{bmatrix} \tag{2.16}$$

The two parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ can now be computed. For the mean, by using the Formula 2.13 and 2.16, we obtain :

$$\sum_{u=1}^{U} \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_u - \hat{\boldsymbol{\mu}}) = 0 \tag{2.17}$$

The maximum-likelihood estimation of the mean $\hat{\boldsymbol{\mu}}$ can be now be deduced:

$$\hat{\boldsymbol{\mu}} = \frac{1}{U} \sum_{u=1}^{U} \mathbf{x}_u \tag{2.18}$$

This result shows that, by using a Normal multivariate distribution, the maximum-likelihood estimate of the mean is equal to the sample mean, i.e. the arithmetic average of the training sample. Considering a cloud of points, the maximum-likelihood estimate of $\boldsymbol{\mu}$ is the centroid.
The same operation can now be applied to the covariance matrix. By using the Formula 2.13 and 2.16 we obtain:

$$\sum_{u=1}^{U} \big(-\frac{1}{2}[\hat{\boldsymbol{\Sigma}}^{-1} + \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}_u - \hat{\boldsymbol{\mu}})(\mathbf{x}_u - \hat{\boldsymbol{\mu}})' \hat{\boldsymbol{\Sigma}}^{-1}] \big) = 0 \tag{2.19}$$

The maximum-likelihood estimation of the covariance matrix $\hat{\boldsymbol{\Sigma}}$ can be deduced:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{U} \sum_{u=1}^{U} (\mathbf{x}_u - \hat{\boldsymbol{\mu}})(\mathbf{x}_u - \hat{\boldsymbol{\mu}})' \tag{2.20}$$

This result shows that, by using a Normal multivariate distribution, the maximum-likelihood estimate of the covariance matrix is the arithmetic average of the U matrices $(\mathbf{x}_u - \hat{\boldsymbol{\mu}})(\mathbf{x}_u - \hat{\boldsymbol{\mu}})'$, i.e. the expected value of the matrix $(\mathbf{x} - \hat{\boldsymbol{\mu}})(\mathbf{x} - \hat{\boldsymbol{\mu}})'$.

2.4 Mixture models

In the previous section we have assumed the probability distribution functions having a multivariate Normal parametric form. For many natural applications, data does not follow a Normal distribution. This is the case when data is formed by several clusters. Using a single Normal distribution could limit the data representativeness. Mixture models are probabilistic models able to represent the presence of sub-populations within an overall population, without using information on the sub-population identity.
Mixture models usually have the following components:

- $\mathcal{D}_{tr} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_U\}$ is the set of feature vectors that we have assumed to be distributed according to the mixture models.
- $\mathcal{Z} = \{z_1, z_2, \ldots, z_U\}$ are the latent variables that indicate to which mixture component each observation belongs, i.e. $z_u \in \{1, 2, \ldots, K\}$. As will be shown in Section 2.4.2, the latent variable could be a likelihood of the mixture occupation.
- $\boldsymbol{w} = \{w_1, w_2, \ldots, w_K\}$ is the vector of weights of the mixtures. They are probabilities between 0 and 1 and the sum of all the weights is equal to 1.
- $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_K\}$ is the set of parameter vectors. For every mixture a vector of parameters is expected to be computed, i.e. the mean and the covariance matrix for a Normal distribution.

Mixture models are defined as:

$$p(\mathbf{x}_u \mid c_m) = \sum_{k=1}^{K} w_k p(\mathbf{x}_u \mid z_u = k, \theta_k) \tag{2.21}$$

Mixture models are typically used for representing complex density functions and the mixture parameters have to be estimated. A common procedure consists in (i) finding clusters in the training data, usually with a K-Means algorithm, (ii) using the centroids as initial mean values of the mixture parameters and (iii) estimating the mixture parameters, usually with the Expectation-Maximization algorithm. At the end of this section the Gaussian Mixture model will be presented as specific case of Mixture models.

2.4.1 K-Means

The K-Means algorithm permits to find sub-populations, i.e. clusters, within a population. A cluster is a group of elements where the elements in the same group are more similar to each other than to those of other groups. Making the hypothesis that the number of clusters K is known, the K-Means algorithm finds groups in which points have a small inter-distance compared to the distances with points belonging to other groups. The inter-distance of a cluster is called within-cluster sum of squares J and it is calculated using the points associated to the centroid of the cluster. The goal of the algorithm is to minimize J and assign every point of the set to a cluster. Considering a data set $\mathcal{D}_{tr}$ of U points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_U\}$, we have:

$$J = \sum_{u=1}^{U} \sum_{k=1}^{K} \zeta_{uk} \|\mathbf{x}_u - \boldsymbol{\mu}_k\| \tag{2.22}$$

where ζ_{uk} is a binary indicator describing to which cluster every point belongs. If the u-th point belongs to the k-th cluster, then $\zeta_{uk} = 1$, otherwise $\zeta_{uk} = 0$. The K-Means algorithm has been extensively used to build lossy coders in signal processing. A given input point $\mathbf{x}_u$ is replaced by its nearest centroid and only the label of the centroid is transmitted over the communication channel. The distorsion by the coder, also called *vector quantizer*, is given by:

$$distorsion(\mathbf{x}_u) = \sum_{k=1}^{K} \zeta_{uk} \|\mathbf{x}_u - \boldsymbol{\mu}_k\| \tag{2.23}$$

and J is the sum of the distorsions over the whole training set.

Input : the data-set ($\mathcal{D}_{tr}$), the number of clusters (K)
Output: the label of each point (ζ_{uk})

initialize the centroids ($\boldsymbol{\mu}^{old}$);
do
  E-step: assign all points to the closest centroid (ζ_{uk});
  M-step: recompute the centroid of each cluster ($\boldsymbol{\mu}^{new}$);
while $\|\boldsymbol{\mu}^{new} - \boldsymbol{\mu}^{old}\| < \epsilon$;
return ζ_{uk}

Algorithm 1: Pseudo-code of the K-Means algorithm.

As shown in the pseudo-code of Algorithm 1, the K-Means algorithm has three steps: (i) initialization, (ii) assignation of all points to the closest centroid and (iii) re-computation of the centroid for each cluster. Steps (ii) and (iii) are also called **E-step** and **M-step**, given the similarity between the K-Means algorithm and the Expectation-Maximization algorithm [3], that will be seen in Section 2.4.2.

Initialization. The first step is usually quite simple. K centroids are randomly initialized from K points in the training set. Finer techniques for the initialization of the centroids exist, for instance by taking random sub-populations and calculating their mean values.

Assignation. In this step all the points are assigned to the closest centroid. Referring to the Formula 2.22, we estimate the ζ_{uk} matrix. If, among all the centroids, the centroid of k-th cluster is the closest to u-th point, then $\zeta_{uk} = 1$, otherwise $\zeta_{uk} = 0$. This can be formalized as:

$$\zeta_{uk} = \begin{cases} 1 & \text{if } k = \text{argmin}_k \ \|\mathbf{x}_u - \boldsymbol{\mu}_k\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.24}$$

Re-computation. When all the points have been assigned to a centroid, the centroid positions are recalculated. By using the Formula 2.22, the derivative of J with the respect to $\boldsymbol{\mu}_k$ is computed and set equal to 0:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 2 \sum_{u=1}^{U} \zeta_{uk}(\mathbf{x}_u - \boldsymbol{\mu}_k) = 0 \tag{2.25}$$

The centroids can be easily computed as:

$$\boldsymbol{\mu}_k = \frac{\sum_u \zeta_{uk} \mathbf{x}_u}{\sum_u \zeta_{uk}} \tag{2.26}$$

The assignation and re-computation steps are iteratively applied until the centroids converge. The convergence is typically determined when the distance between the new and previous values of the centroids is below a small value ϵ. At the end of the K-Means algorithm, the centroid positions are returned with the information to which cluster every point is assigned (ζ_{uk}).

In Figure 2.4 an example of the K-Means with $K = 3$ is shown. Several iterations are performed from the initial position (iteration 1), where the centroids are randomly positioned, until the last position (iteration 14), where the centroids converge to their final position. Other strategies for the K-Means algorithm do exist, such as the binary splitting, where K is a power of 2 and where, after the convergence of several assignation and re-computation steps, the centroids are split into 2 new centroids, e.g. giving them a small perturbation. By using the K-Means algorithm, the latent variables $\mathcal{Z}$ are initialized simply by using the Formula:

$$z_u = \underset{k}{\text{argmax}} \ \zeta_{uk} \tag{2.27}$$

This initialization will be used as starting point for the Expectation-Maximization algorithm, as explained in the next section.

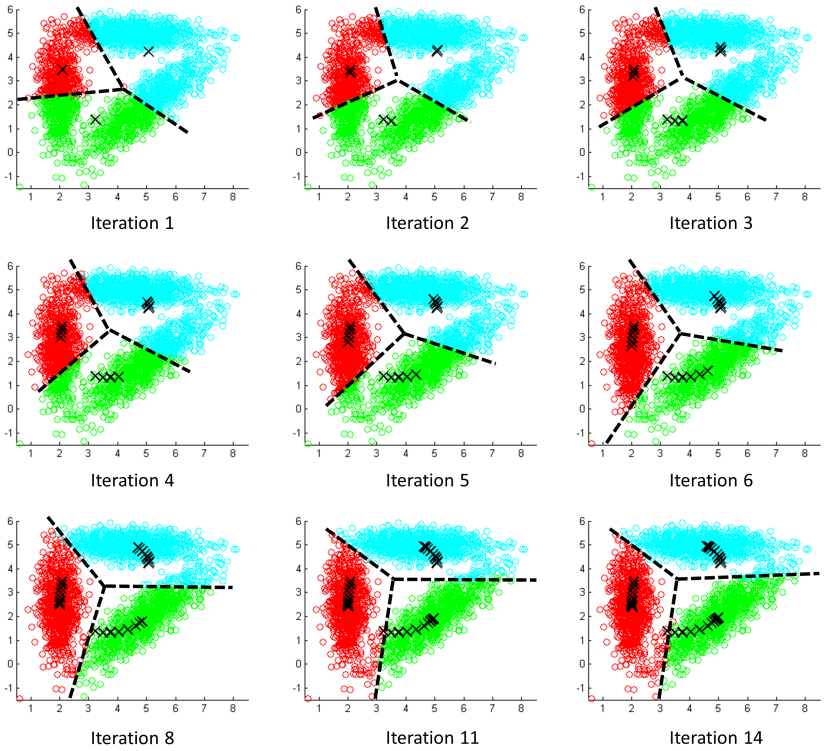


Figure 2.4: Example of assignation and re-computation steps of the K-Means algorithm. The centroids (black cross) are randomly positioned at the iteration 1. K-Means iterations of the assignation and re-computation steps are then applied until iteration 14 when the centroids are stabilizing. The dotted lines represent the Voronoi regions.

2.4.2 Expectation-Maximization algorithm

By using the K-Means algorithm we are able to divide the population into K clusters. These clusters are often used to initialize the parameters in a Mixture model. After the computation of the parameters, the probability of every point to belong to every mixture is computed. The main idea of the Expectation-Maximization algorithm is to iteratively alternate these two steps, called **E-step** and **M-step** [17]. The EM algorithm is used to complete the observed data with the latent information $\mathcal{Z}$ about the likelihood of the mixture occupations.

Usually in literature the expected log-likelihood function $\ln p(\mathbf{x}_u \mid \theta)$ is called Q-function (or **auxiliary function**). It is written as $Q(\theta, \theta^{old})$, in the sense that the likelihood variation from θ^{old} to θ is always greater than $Q(\theta, \theta^{old})$. The auxiliary function can be written as:

$$Q(\theta, \theta^{old}) \equiv E\left[\sum_{u=1}^{U} \ln p(\mathbf{x}_u, z_u \mid \theta)\right] \tag{2.28}$$

Therefore:

$$Q(\theta, \theta^{old}) = \sum_{u=1}^{U} E\left[\ln\left[\prod_{k=1}^{k} w_k p(\mathbf{x}_u, z_u \mid \theta_k)\right]\right] \tag{2.29}$$

The formula can be written as:

$$Q(\theta, \theta^{old}) = \sum_{u=1}^{U}\sum_{k=1}^{K} r_{uk} \ln[w_k p(\mathbf{x}_u \mid \theta_k)] \tag{2.30}$$

where $r_{uk} = p(z_u = k, \mathbf{x}_u \mid \theta_k^{(t-1)})$ is the **responsibility** taken by the k-th mixture for the u-th point.

Then:

$$Q(\theta, \theta^{old}) = \sum_{u=1}^{U}\sum_{k=1}^{K} r_{uk} \ln w_k + \sum_{u=1}^{U}\sum_{k=1}^{K} r_{uk} p(\mathbf{x}_u \mid \theta_k) \tag{2.31}$$

Input : the data-set ($\mathcal{D}_{tr}$), the number of mixtures (K)
Output: the vector of parameters ($\boldsymbol{\theta}$)
Initialization of the parameters ($\boldsymbol{\theta}^{old}$);
do
  E-step: compute the responsibility taken by every mixture for every point (r_{uk});
  M-step: recompute the parameters ($\boldsymbol{\theta}$);
while $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) < \epsilon$;
return $\boldsymbol{\theta}$

Algorithm 2: Pseudo-code of the EM algorithm.

The EM algorithm has three steps, as detailed in the pseudo-code of the Algorithm 2.
Initialization. The population is divided in clusters as starting point. Usually a K-Means is used for determining the initial ζ_{uk}. However, other procedures exist: for instance data could be randomly separated [18]. The parameters $\boldsymbol{\theta}^{old}$ are computed for every mixture starting by ζ_{uk}.
E-step. For every mixture k the responsibility r_{uk} is computed:

$$r_{uk} = \frac{w_k p(\mathbf{x}_u \mid \theta_k^{old})}{\sum_k w_k p(\mathbf{x}_u \mid \theta_k^{old})} \tag{2.32}$$

M-step. During the M-step, the Q-function is optimized with the respect of w_k and θ_k. The mixture weight w_k can be easily computed as:

$$w_k = \frac{1}{U}\sum_{u=1}^{U} r_{uk} \tag{2.33}$$

The algorithm stops when the parameters do not change or when other convergence conditions are satisfied. The optimization of the parameters depends on the form of the probability density function used at the right hand side of the Formula 2.31. In the next section we develop the reasoning for Gaussian Mixture models.

2.4.3 Gaussian Mixture models

For Gaussian Mixture models (GMMs), Normal distributions are used for representing the probability densities. In the multivariate Normal case, the parameter vector θ is $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix. Here the probability densities are computed as a weighted sum of multivariate Normal densities and the parameter vector θ is $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K, w_1, \ldots, w_K\}$, with:

$$p(\mathbf{x}_u \mid \theta) = \sum_{k=1}^{K} \frac{w_k}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_k|}} exp[-\frac{1}{2}(\mathbf{x}_u - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_u - \boldsymbol{\mu}_k)] \tag{2.34}$$

The estimation of the parameters is done in the M-step. The maximum-likelihood estimation of the mean and the covariance can be computed as previously shown in Section 2.3.1. In the specific case, for the mean we obtain:

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_u r_{uk} \mathbf{x}_u}{\sum_u r_{uk}} \tag{2.35}$$

This is a satisfying result, because the mean of the k-th mixture is the weighted average of all the points assigned to the mixture. For the covariance we obtain:

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_u r_{uk} (\mathbf{x}_u - \boldsymbol{\mu}_k)(\mathbf{x}_u - \boldsymbol{\mu}_k)'}{\sum_u r_{uk}} \tag{2.36}$$

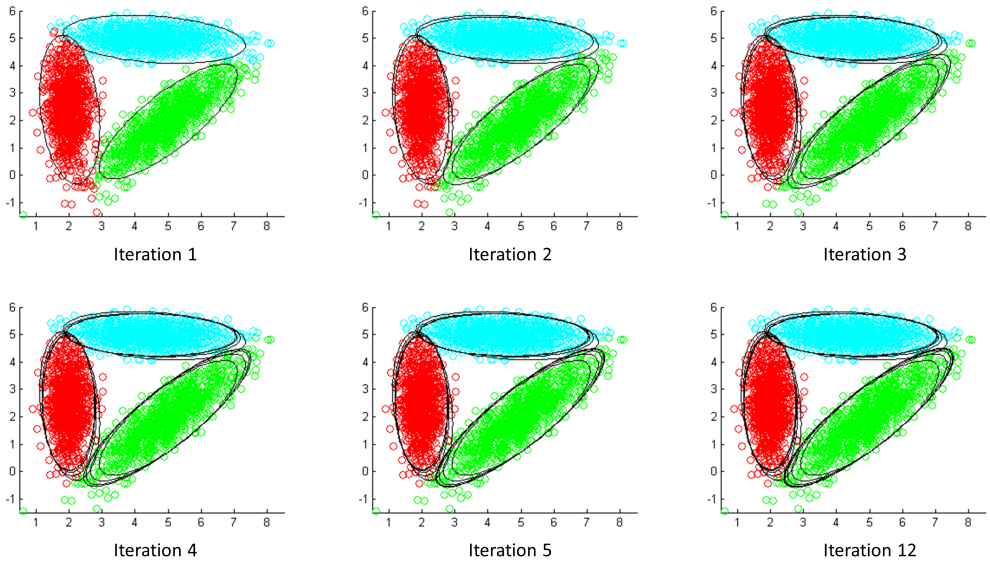


Figure 2.5: Example of the EM algorithm applied to GMMs. The ellipses have been centered on the centroids and the axes are 3 times the standard deviations. At the first iteration, we use the labels of the points obtained after the K-Means. After 12 iterations, the algorithm converges.

This is also a satisfying result, because the covariance is proportional to the weighted empirical scatter matrix. Figure 2.5 gives an example of the EM algorithm with 3 Gaussians. The starting iteration uses the labels of the points obtained after the K-Means.

The hypothesis that the feature coefficients are uncorrelated is made in several implementations. As a consequence, we can simplify the density computation by using diagonal covariance matrices. When the components are acting together to model the overall probability density functions, the full covariance matrices are not necessary. The linear combination of diagonal covariance components is capable of modeling the correlations between feature vector elements even if the features are not statistically independent. In addition the effect of using a set of full covariance components is the same that using a larger set of diagonal covariance components [19].

2.5 Hidden Markov models

The GMMs previously presented do not include the notion of state. In many cases we deal with dynamic systems that change states as a function of time and, by taking into account this information, modeling performances could be improved. By collecting samples from a dynamic system over a time interval, we obtain a time series of new observations $\mathbf{O} = \{\mathbf{o}_t : t \in \tau\}$ where $\tau = \{1, 2, \ldots, T\}$ is the index set. As explained in the introduction of the chapter, usually feature vectors are computed from the observations. Coherently with the notation previously introduced, we note the time series of features as $\mathbf{X} = \{\mathbf{x}_t : t \in \tau\}$. For sake of simplicity, the time series $\mathbf{X}$ will also be called observation sequence.

Algorithms as Hidden Markov models (HMMs) are able to deal with *time series* generated by systems having different "states of emission". HMMs are able to take several decisions during time where the current decision depends on the previous one. In other words HMMs are a Markov process where the future does not depend on the past. In fact, the conditional probability distribution of future states of the process depends only upon the present state.

A Markov model is described by a set of states $\mathcal{S} = \{S_1, S_2, ..., S_N\}$, where N is the number of states. The probabilities of transitioning from one state to another are called *transition probabilities* and the conditional distributions of observations given states are called *emission probabilities*. Usually two non-emitting states, i.e. states with no probability distribution associated, are appended to the

set of states [20]:

- S_{start} is the start state, also called non-emitting initial state. It does not have self-transition, only transitions to other states. Considering a time series, this state represents the time just before the first observation.

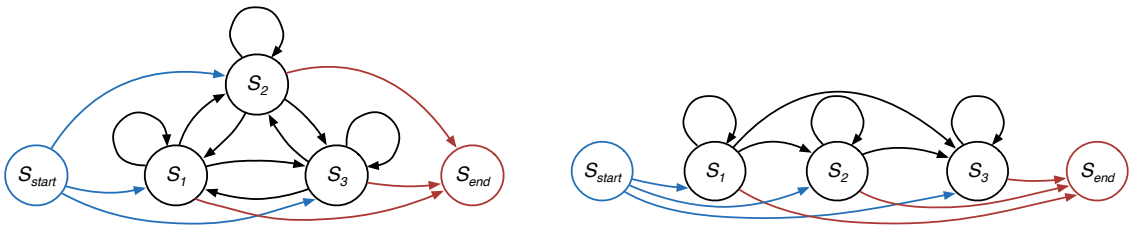
- S_{end} is the end state, also called non-emitting final state or *absorbing* state. It does not have self-transition, only the transitions from the other states. Considering a time series, this state represents the time just after the last observation.

As a consequence, the set of states can also be represented as $\mathcal{S} = \{S_{start}, S_1, \ldots, S_N, S_{end}\}$. How the states are connected determines the topology of the model. The most used topologies are:

1. *Ergodic*. Every state is connected to all the others. In one step any other state can be reached, as illustrated in Figure 2.6a.

2. *Forward* (or *left-right*). Every state is connected to itself and to the ones having a higher index. States already seen in the past can not be visited again, as illustrated in Figure 2.6b. A particular case of the *forward* category is the *forward single-step*, where only the self-transition and the transition to the next state are permitted.

3. *Specific topology*. Other topologies are used by taking advantage of the available a priori information between state connections.



(a) Ergodic topology (b) Forward topology

Figure 2.6: Example of the ergodic and forward topologies by using 3 states.

Ergodic topologies are popular in some fields where any jump from one state to the other is possible or when the *transition patterns* are too variable and poorly discriminant. Also, to some extent, an ergodic HMM with equiprobable transitions is similar in nature to a GMM where each mixture would correspond to one state. Ergodic systems have been used in different fields such as activity recognition [21], language independent character recognition [22] and font size and type identification in text [23][24]. Some other fields usually deal with temporal or spatial signals characterized by a specific sequence of states for which a forward or specific topology is more popular. In this case, the topology incorporates a priori knowledge of allowed sequences of states, representing for example the semantic of the modeled observations. The mainstream example is in speech recognition where words are usually modeled as a specific forward sequence of phonemes [7].

More formally, a HMM λ is completely defined when the transition matrix A, the emission matrix B and the initial state probabilities π are defined. A HMM is actually modeling 2 stochastic processes, the first one is bound to the *transitions* between states and the second one to the *emissions* of observations in the states. In HMMs, the term hidden is referring to the fact that the state sequence is usually not known.

Transition probabilities. The relations between states are expressed in terms of transition probabilities. A is the transition matrix, arranged as a $N \times N$ matrix:

$$A = \{a_{ij}\} = \{p(q_{t+1} = S_j | q_t = S_i])\} \tag{2.37}$$

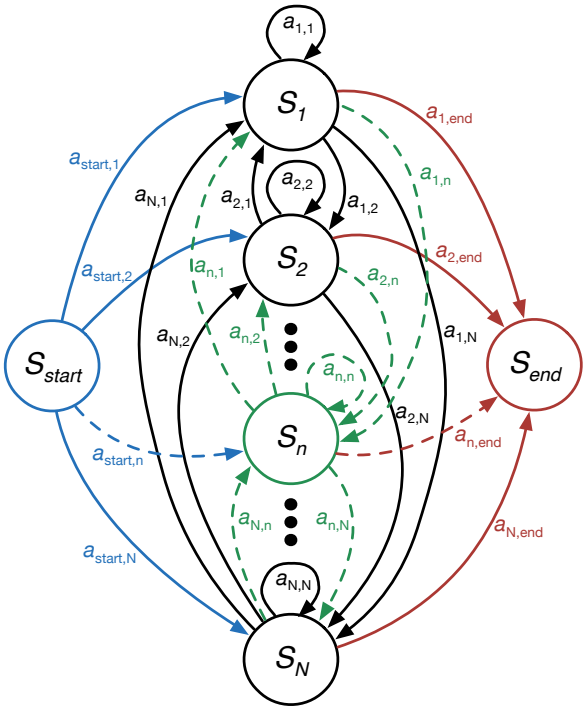


Figure 2.7: Example of a generic HMM with a set of states having $\mathcal{S} = \{S_{start}, S_1, \ldots, S_N, S_{end}\}$. The transition values are reported on the arrows.

where $1 \leq i, j \leq N$ and q_t is the state at time t. Considering the set of states $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, the transition matrix has the following form:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{iN} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nj} & \cdots & a_{NN} \end{pmatrix} \tag{2.38}$$

A property of the transition matrix is that the sum of the elements on the i-th row has to be 1, i.e. the sum of the outgoing transition probabilities of the state is 1, as shown in Formula 2.2. The diagonal of the transition matrix represents the self-transitions. In the case of a *forward* topology, all the elements below the diagonal are equal to 0. An example of a generic HMM with the indication of the transition values is shown in Figure 2.7.

Emission probabilities. The conditional probability distributions of the observed variables for each state are reported in a set of N probability density functions tied to the states. The emission probabilities representation depends on the type of observations. For *discrete observations* with $\mathbf{o}_t \in \boldsymbol{\eta} = \{\eta_1, \eta_2, \ldots, \eta_H\}$, where $\boldsymbol{\eta}$ is the vector of symbols, the emission probabilities can be collected in the emission matrix B:

$$B = \{b_j(\mathbf{x}_t)\} = \{p(\mathbf{x}_t = \eta_h | q_t = S_j)\} \tag{2.39}$$

where $1 \leq j \leq N$.

For *continuous observations* with $\mathbf{x}_t \in \mathbb{R}^D$ the emission probabilities can be collected in a vector of functions B:

$$B = \{b_j(\mathbf{x}_t)\} = \{p(\mathbf{x}_t | q_t = S_j)\} \tag{2.40}$$

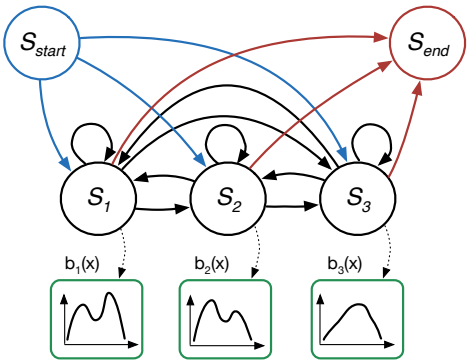


Figure 2.8: Example of an ergodic HMM with 3 states. The emission probability density functions for each state are illustrated assuming a mono-dimensional input space.

where $1 \leq j \leq N$ and $b_j(\mathbf{x}_t)$ is a function of the observations, often parametrized by mixture of Gaussians.

Considering as probability distribution a Gaussian Mixture model with K components, we can write:

$$b_j(\mathbf{x}_t) = \sum_{k=1}^{K} w_{jk} \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}^{jk}, \boldsymbol{\Sigma}^{jk}) \tag{2.41}$$

As stated in Section 2.4.2, if the hypothesis of uncorrelated feature coefficients is made, the covariance matrices are diagonal and therefore $\boldsymbol{\mu}^j$ and $\boldsymbol{\Sigma}^j$ can be written in the following form:

$$\boldsymbol{\mu}^j = \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1k} & \cdots & \mu_{1K} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2k} & \cdots & \mu_{2K} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mu_{d1} & \mu_{d2} & \cdots & \mu_{dk} & \cdots & \mu_{dK} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mu_{D1} & \mu_{D2} & \cdots & \mu_{Dk} & \cdots & \mu_{DK} \end{pmatrix} \tag{2.42}$$

$$\boldsymbol{\Sigma}^j = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} & \cdots & \sigma_{1K} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} & \cdots & \sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dk} & \cdots & \sigma_{dK} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \sigma_{D1} & \sigma_{D2} & \cdots & \sigma_{Dk} & \cdots & \sigma_{DK} \end{pmatrix} \tag{2.43}$$

where D is the dimension of the feature space. The weights of the Gaussians Mixture models are noted:

$$\boldsymbol{w}^j = \begin{pmatrix} w_1 & w_2 & \cdots & w_k & \cdots & w_K \end{pmatrix} \tag{2.44}$$

In conclusion the emission matrix B can be completely described by the matrices $\boldsymbol{\mu}^j$, $\boldsymbol{\Sigma}^j$ and $\boldsymbol{w}^j$, assuming that the probability density functions are represented by GMMs. In Figure 2.8 we show an example of an ergodic HMM with 3 states with a representation of the density probability functions.

Initial state probabilities. π is the set of initial state probabilities of being in state q at time $t = 0$:

$$\pi = \{\pi_i\} = \{p(q_1 = S_i)\} \tag{2.45}$$

where $1 \leq i \leq N$. The initial state probabilities π could also be included in the transition matrix with:

$$\pi_i = p(q_1 = S_i \mid q_0 = S_{start}) = a_{0i} \tag{2.46}$$

As stated in [14], three canonical problems are usually faced:

1. Given a HMM λ and an observation sequence, we compute the probability that the sequence has been generated by the model. This problem, called **evaluation problem**, is usually solved by using the forward evaluation of the Baum-Welch algorithm.

2. Given an observation sequence and a model λ, we can "decode" the observation sequence by recovering the most probable sequence of hidden states. This problem, called **decoding problem**, is usually solved by using the Viterbi algorithm.

3. Given a set of training observations and the model topology, we can compute the parameters of the model that best explains the statistics of the observations. This problem, called **learning problem**, is usually solved using an EM procedure applied using either a Viterbi or Baum-Welch criterion.

2.5.1 Evaluation problem

In this problem the model $\lambda(A, B, \pi)$ and an observation sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ are given. We want to compute the probability that the specific sequence $\mathbf{X}$ has been generated by the model. The emission probability of the observation sequence $\mathbf{X}$ given a specific state sequence $Q = \{q_1, q_2, \ldots, q_T\}$ is:

$$p(\mathbf{X} \mid Q, \lambda) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid q_t, \lambda) = \prod_{t=1}^{T} b_j(\mathbf{x}_t | q_t = S_j) \tag{2.47}$$

The transition probability of the state sequence Q is:

$$p(Q \mid \lambda) = \pi_{q_1} \prod_{t=1}^{T} p(q_{t+1} = S_j | q_t = S_i) = \pi_{q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} \tag{2.48}$$

The joint probability of $\mathbf{X}$ and q given the model is therefore:

$$p(\mathbf{X}, Q \mid \lambda) = p(\mathbf{X} \mid Q, \lambda)\ p(Q \mid \lambda) = \pi_{q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} b_j(\mathbf{x}_t) \tag{2.49}$$

And the probability of $\mathbf{X}$ given the model λ is simply the sum of all the joint probabilities over the set of possible sequence of states:

$$p(\mathbf{X} \mid \lambda) = \sum_{allpath} p(\mathbf{X}, Q \mid \lambda) = \sum_{allpath} \pi_{q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} b_j(\mathbf{x}_t) \tag{2.50}$$

where "*all path*" is the number of all possible sequences of hidden states. Emitting all the paths and computing 2.50 is often not feasible in practice due to the complexity $O(N^T T)$. The *Forward-Backward* algorithm proposes an alternative solution. For every time step, we can recursively compute a list of $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_T\}$ where $\alpha_t = [\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(i), \ldots, \alpha_t(N)]$ and where $\alpha_t(i)$ is the probability of generating the first t elements at the i-th state:

$$\alpha_t(i) = p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, q_t = S_i \mid \lambda) \tag{2.51}$$

The α is initialized for $t = 1$ with:

$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1) \tag{2.52}$$

As shown in Figure 2.9, we can calculate for the t-th step:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(\mathbf{x}_{t+1}) \tag{2.53}$$

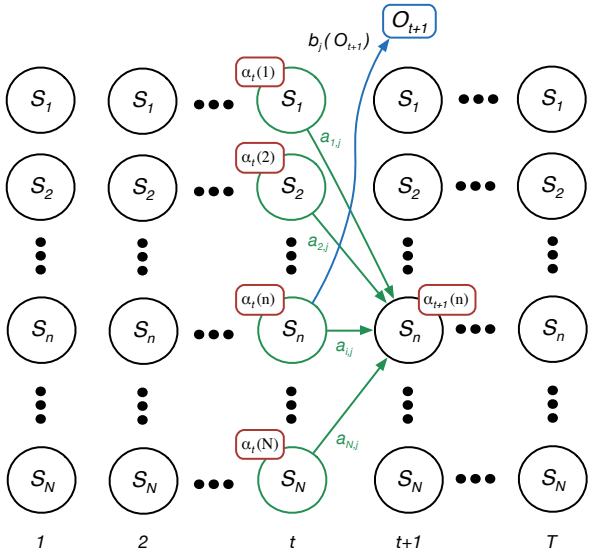


Figure 2.9: Visualization of the forward step. The α are represented in red, the observation in blue and the states and transitions at the time t in green.

The complexity of the Forward algorithm is $O(N^2T)$. The pseudo-code is presented in Algorithm 3.

Input : the model $\lambda(A, B, \pi)$; the observed sequence $(\mathbf{X})$
Output: the probability of the observed sequence $p(\mathbf{X} \mid \lambda)$

for $i \leftarrow 1$ **to** N **do** // Initialization
  $\alpha_1(i) = \pi_i b_i(\mathbf{x}_1)$;
end
for $t \leftarrow 1$ **to** $T-1$ **do** // Recursion
  for $j \leftarrow 1$ **to** N **do**
    $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{x}_{t+1})$;
  end
end
return $\sum_{i=1}^{N} \alpha_T(i)$

Algorithm 3: Pseudo-code of the Forward algorithm.

The Backward algorithm is not strictly necessary for solving the **evaluation problem**, anyway, given that it will be used for the **learning problem**, it is here introduced. The procedure is similar to the Forward algorithm, but reversed in time. For every step we can recursively compute a list of $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_T\}$ where $\beta_t = [\beta_t(1), \beta_t(2), \ldots, \beta_t(i), \ldots, \beta_t(N)]$ and where $\beta_t(i)$ is the probability of generating the last elements from $t+1$ to the end at the i-th state:

$$\beta_t(i) = p(\mathbf{x}_{t+1}\ \mathbf{x}_{t+2}\ \ldots\ \mathbf{x}_T \mid q_t = S_i, \lambda) \tag{2.54}$$

The β is initialized for $t = T$ with:

$$\beta_T(i) = 1 \tag{2.55}$$

The backward probabilities are computed in a similar manner as for the forward procedure:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j) \tag{2.56}$$

The pseudo-code is presented in Algorithm 4.

Input : the model $\lambda(A, B, \pi)$; the observed sequence $(\mathbf{X})$
Output: the backward probabilities (β)

Initialization:
for $i \leftarrow 1$ **to** N **do** // Initialization
  $\beta_T(i) = 1$;
end
for $t \leftarrow T - 1$ **to** 1 **do** // Recursion
  for $i \leftarrow 1$ **to** N **do**
    $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j)$;
  end
end
return β

Algorithm 4: Pseudo-code of the Backward algorithm.

2.5.2 Decoding problem

While in the previous problem we found the probability of a sequence $\mathbf{X}$ to be observed given the model, in the decoding problem instead we are interested in finding the most probable sequence of states for a given observation. As in the previous case, this problem could be solved by enumerating all the possible paths and choosing the most probable one. However this solution is very often not tractable due to the high complexity in $O(N^T T)$. The Viterbi algorithm proposes an alternative solution able to reduce the complexity to $O(N^2 T)$. We define a new variable $\delta_t(j)$ as the highest probability along a single path and the relatives arrays of paths $\psi_t(j)$. In practice we keep in memory at the t-th step for every state the most probable state and the relative probability. First of all, we have to initialize the δ and ψ variables:

$$\begin{cases} \delta_1(i) = \pi_i b_i(\mathbf{x}_1) \\ \psi_1(i) = 0 \end{cases} \tag{2.57}$$

In the following step we can compute the $\delta_t(j)$ as:

$$\delta_t(j) = \max_i \; [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{x}_t) \tag{2.58}$$

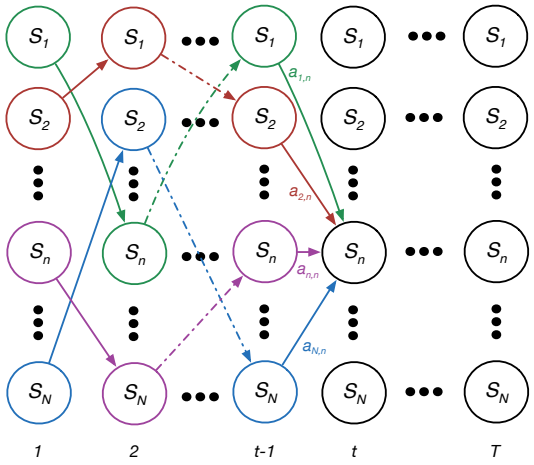


Figure 2.10: Example of the Viterbi algorithm: for every state at the time $t - 1$ for the generic state S_n a different path is selected (indicated by different colors).

and $\psi_t(j)$ as:

$$\psi_t(j) = \underset{i}{\text{argmax}} \ [\delta_{t-1}(i)a_{ij}] \tag{2.59}$$

By using the ψ variable, we are able to retrieve the most probable sequence of hidden states, starting to the last and coming back to the first one (backtracking step):

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \tag{2.60}$$

The Viterbi algorithm is rather similar to the forward procedure and in fact they have the same complexity. Instead of summing the probabilities of the possible paths, the Viterbi algorithm actually chooses the most probable path. The pseudo-code is presented in Algorithm 5. In Figure 2.10 we show an example of the decoding problem.

Input : the model $\lambda(A, B, \pi)$; the observed sequence $(\mathbf{X})$
Output: the sequence of the most probable states (q)

for $i \leftarrow 1$ **to** N **do** `// Initialization`
  $\delta_1(i) = \pi_i b_i(\mathbf{x}_1)$;
  $\psi_1(i) = 0$;
end
for $t \leftarrow 2$ **to** T **do** `// Recursion`
  for $j \leftarrow 1$ **to** N **do**
    $\delta_t(j) = \max_i[\delta_{t-1}(i)a_{ij}]b_j(\mathbf{x}_t)$;
    $\psi_t(j) = \text{argmax}_i \ [\delta_{t-1}(i)a_{ij}]$;
  end
end
$Q_T = \text{argmax}_i \ [\delta_t(i)]$;
for $t \leftarrow (T-1)$ **to** 1 **do** `// Backtracking`
  $q_t^* = \psi_{t+1}(q_{t+1}^*)$;
end
return q

Algorithm 5: Pseudo-code of the Viterbi algorithm.

2.5.3 Learning problem

The learning problem is the most difficult among the three canonical problems. In this case we have a set of observation data, namely a training set, and we want to learn the model parameters. Two approaches can be used for the training: (i) the *Baum-Welch criterion* and the *Viterbi criterion*.

Baum-Welch criterion

The main idea is to calculate an estimation of the transition matrix $\hat{A}$ and of the emission matrix $\hat{B}$ and iteratively improve their estimation up to their convergence. We can reuse the definition of the α and β probabilities of Formula 2.51 and 2.54. $\alpha_t(i)$ is the probability of being in state i and generating the sequence from the start to the step t. $\beta_t(i)$ is the probability of being in state i and generating the sequence from the end back to the step $t+1$. For this problem the actual values of A and B are not available, therefore we can compute only an estimation of the α and β parameters.

It is useful to define two new variables: $\gamma_t(i)$ representing the probability of being at the state S_i at time t given the observation $\mathbf{X}$ and $\xi_t(i, j)$ representing the probabilities of being in the state S_i at the time t and at the state S_j at the time $t+1$:

$$\gamma_t(i) \equiv p(q_t = S_i \mid \mathbf{X}, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{p(\mathbf{X} \mid \lambda)} \tag{2.61}$$

$$\xi_t(i, j) \equiv p(q_t = S_i, q_{t+1} = S_j \mid \mathbf{X}, \lambda) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{x}_{t+1})\beta_{t+1}(j)}{p(\mathbf{X} \mid \lambda)} \tag{2.62}$$

The two variables can be explicitly related:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.63}$$

We can now re-estimate the parameters of λ:

- $\hat{\pi}_i$ is the expected number of times in the state S_i at the time $t = 1$, therefore $\hat{\pi}_i = \gamma_1(i)$.
- $\hat{a}_{ij}$ is equal to the expected number of transitions from the state S_i to the state S_j over the expected number of transitions from the state S_i. Therefore:

$$\hat{a}_{ij} = \frac{\sum_t \xi_t(i,j)}{\sum_t \gamma_t(i)} \tag{2.64}$$

- $\hat{b}_j(k)$ for GMMs with K mixtures can be rewritten in terms of $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{w}$ [25]:

$$\hat{\boldsymbol{\mu}}_k^j = \frac{\sum_t \gamma_t(j,k)\mathbf{x}_t}{\sum_t \gamma_t(j,k)} \tag{2.65}$$

$$\hat{\boldsymbol{\Sigma}}_k^j = \frac{\sum_t \gamma_t(j,k)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_k^j)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_k^j)'}{\sum_t \gamma_t(j,k)} \tag{2.66}$$

$$\hat{\boldsymbol{w}}_k^j = \frac{\sum_t \gamma_t(j,k)}{\sum_t \gamma_t(j)} \tag{2.67}$$

  where $\gamma_t(j)$ is described by:

$$\gamma_t(j) = \sum_{k=1}^{K} \gamma_t(j,m) \tag{2.68}$$

  and $\gamma_t(j,k)$ is the *mixture-occupation likelihood* and can be computed by:

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_j \alpha_t(j)\beta_t(j)}\right]\left[\frac{\boldsymbol{w}_k^j \mathcal{N}(\mathbf{x}_t, \boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j)}{\sum_k \boldsymbol{w}_k^j \mathcal{N}(\mathbf{x}_t, \boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j)}\right] \tag{2.69}$$

We can now consider the new model $\hat{\lambda}(\hat{A}, \hat{B}, \hat{\pi})$ for finding the most probable sequence of hidden states for every observation. This process is iteratively done up to a convergence condition is realized. The Baum-Welch algorithm can be considered as an Expectation-Maximization algorithm. As the EM algorithm, Baum-Welch is guaranteed to converge in a local maximum.

Viterbi criterion

An alternative solution for the learning problem is by using the *Viterbi criterion*. The main idea is to re-estimate HMMs parameters by maximizing the probability of the best HMM state sequence for each training sample given the model. This solution is also called *segmental k-means* [26]. Compared to the Baum-Welch criterion, it requires less computational cost for somehow similar performances if enough training material is available. For this reason, the Viterbi criterion is a common choice among researchers. For the reason mentioned before, Viterbi criterion works well with large amount of data.

The Viterbi training starts from an initial model λ. The algorithm is based on two steps: (i) segmentation and (ii) optimization. The segmentation consists in decoding the most probable sequence of states, as efficiently performed by Viterbi algorithm. The optimization step finds a new set of parameters $\hat{\lambda}$ optimizing the likelihoods. Given a state sequence Q and the observation $\mathbf{X}$, this can be written as:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \ \{\max_S \ p(\mathbf{X}, Q \mid \lambda)\} \tag{2.70}$$

Now we are able to replace the original model λ with the new $\hat{\lambda}$. This operation can be iterated until the state-optimized likelihood converges. The maximization of the state-optimized likelihood

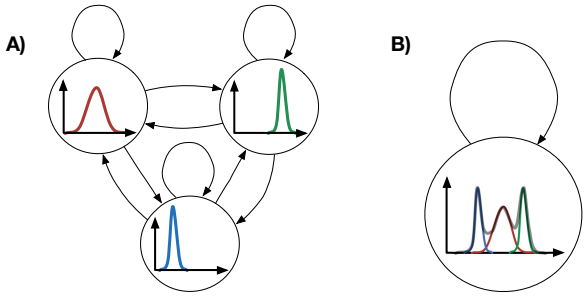


Figure 2.11: HMM with three states (a) an its equivalent of one-state (b) obtained by merging the probability density functions.

has to be done for every state in S. For the GMM case, the emission probabilities can be computed as follows:

$$\hat{\boldsymbol{\mu}}_k^j = \frac{\sum_t \xi_t(j,k)\mathbf{x}_t}{\sum_t \xi_t(j,k)} \tag{2.71}$$

$$\hat{\boldsymbol{\Sigma}}_k^j = \frac{\sum_t \xi_t(j,k)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_k^j)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_k^j)'}{\sum_t \xi_t(j,k)} \tag{2.72}$$

$$\hat{\boldsymbol{w}}_k^j = \frac{\sum_t \xi_t(j,k)}{\sum_t \sum_k \xi_t(j,k)} \tag{2.73}$$

where $\xi_t(j,k)$ is a binary indicator function:

$$\xi_t(j,k) = \begin{cases} 1 & \text{if } k = \text{argmin}_k \ \boldsymbol{w}_k^j \mathcal{N}(\mathbf{x}_t, \boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j) \\ 0 & \text{otherwise.} \end{cases} \tag{2.74}$$

Rodriguez et al. [27] compared the performances of the *Baum-Welch criterion* and the *Viterbi criterion* by using discrete and continuous HMMs for a task in the speech recognition field. When using discrete HMMs, Viterbi criterion yields in some cases suboptimal models while the Baum-Welch criterion is more robust. When using continuous HMMs, Viterbi criterion is as good as Baum-Welch criterion with the advantage of much less computational efforts.

2.5.4 One-state HMM

A particular type of HMM is the so-called *one-state HMM*, representing the degenerated case. Such a model has only a self transition. The Viterbi algorithm is not used given that the sequence of states for each observation is known a priori. The one-state HMM has no more "hidden" information, therefore the model is uniquely based on the estimation of the probability density functions. While a standard HMM is constituted by a set of states, a one-state HMM could be seen as stateless. If the probability density functions are approximated with mixtures, the one-state HMM is equivalent to a Mixture model. In addition, if the one-state HMM is described by Mixture of Gaussians, the model is equivalent to a GMM.

HMMs are useful when observations can be represented by different probability density functions depending on particular temporal segments. When the observations do not have this difference between temporal segments, the estimation of the probability density functions can be merged and used in a HMM with one-state. An example is illustrated in Figure 2.11: a HMM with three states (a) that has an estimation of the probability density function of a single Gaussian per state is equivalent to a one-state HMM with the three Gaussians merged together (b).

2.5.5 HMM hypotheses

Hereunder we resume the principal hypotheses made when using a HMM [28, 15]:

- *The time series are assumed to be produced by a HMM λ built from a set of N states $S = \{S_1, S_2, \ldots, S_N\}$.* In general in this thesis we use an ergodic topology, otherwise it will be explicitly indicated.

- *First order Markov process, i.e. independence property of the hidden process.* The hypothesis of a Markov process implies that the next step depends on the current state and not on the previous states. For this reason the Markov process is also called "memoryless".

- *Observation independence. A sub-sequence of observations $\{x_{i-d}, \ldots, x_i, \ldots x_{i+d}\}$ depends on the state visited at the time i.* In order to simplify the problem, usually d is equal to 0. This means that what is happening in the state visited at time i does not depend on the past and future samples. However, indirect information about the next and previous samples can be taken into account with other methods, for instance with the dynamic coefficients, as we will explain in the next chapter.

- *The current observation x_n at time n only depends on the current state.* The emission probabilities are computed only using the current observation. This means that the emission probabilities are related to states instead of the transitions. This hypothesis reduces the number of probability density functions, given that the number of transitions is generally more than the number of states.

- *Independence of the features.* A hypothesis that is commonly made is the assumption of independence between features. This hypothesis implies to work with diagonal matrices instead of full covariance matrices, providing a great speed up of the computational time. Even if this hypothesis is usually not completely true, it is commonly used for reducing the computational time and the complexity of the system in terms of number of parameters.

Bibliography

[1] A. Ng, "Coursera: Stanford University's Machine Learning course." http://online.stanford.edu/course/machine-learning-1, 2014.

[2] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley & Sons, 2000.

[3] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 4. Springer Science+Business Media, LLC, 2006.

[4] D. MacMillen, "An industrial view of electronic design automation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1428–1447, 2000.

[5] R. Probert and K. Saleh, "Synthesis of communication protocols: Survey and assessment," *IEEE Transactions on Computers*, vol. 40, no. 4, pp. 468–476, 1991.

[6] E. Kaiser, M. Johnston, and P. Heeman, "PROFER: predictive, robust finite-state parsing for spoken language," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1999.

[7] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, vol. 103. Prentice-Hall, Inc, 1993.

[8] R. Plamondon and S. Srihari, "On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[9] Y. Zhang, "Prediction of Financial Time Series with Hidden Markov Models," *The Magazine of the Fine Arts*, vol. B. Eng, no. May, p. 102, 2004.

[10] K. Androutsopoulos, D. Clark, M. Harman, J. Krinke, and L. Tratt, "State-Based Model Slicing : A Survey," *ACM Computing Surveys*, vol. 1, no. 212, 2013.

[11] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes," *Advances in Neural Information Processing Systems*, vol. 14, pp. 605–610, 2001.

[12] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. The MIT Press, 2006.

[13] P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 1991.

[14] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[15] J. Hennebert, *Hidden Markov models and artificial neural networks for speech and speaker recognition*. PhD thesis, EPFL Lausanne, №1860, 1998.

[16] D. Reynolds, "Gaussian Mixture Models," *Encyclopedia of Biometric Recognition*, 2008.

[17] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM Algorithm," *Journal of Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[18] V. Melnykov and I. Melnykov, "Initializing the EM algorithm in Gaussian mixture models with an unknown number of components," *Computational Statistics & Data Analysis*, vol. 56, no. 6, pp. 1381–1395, 2012.

[19] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[20] R. Singh, *Sphinx-III recognition system.*

[21] E. Kim, S. Helal, and D. Cook, "Human Activity Recognition and Pattern Discovery," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 48–53, 2010.

[22] F. Einsele, R. Ingold, and J. Hennebert, "A Language-Independent, Open-Vocabulary System Based on HMMs for Recognition of Ultra Low Resolution Words," *Journal of Universal Computer Science*, vol. 14, pp. 2982–2997, October 2008.

[23] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, "A Study on Font-Family and Font-Size Recognition Applied to Arabic Word Images at Ultra-Low Resolution," *Pattern recognition Letters (PRL)*, vol. 34, pp. 209–218, January 2013.

[24] F. Slimane, S. Kanoun, A. Alimi, R. Ingold, and J. Hennebert, "Gaussian Mixture Models for Arabic Font Recognition," in *20th International Conference on Pattern Recognition, ICPR 2010*, (Istanbul (Turkey)), pp. 2174–2177, aug 2010.

[25] P. Jackson, "HMM tutorial." http://www.ee.surrey.ac.uk/Teaching/Courses/eem.ssr/, 2011.

[26] B. Juang and L. Rabiner, "The segmental K-means algorithm for estimating parameters of Hidden Markov Models," 1990.

[27] L. Rodríguez and I. Torres, "Comparative Study of the Baum-Welch and Viterbi Training Algorithms Applied to Read and Spontaneous Speech Recognition," *Pattern Recognition and Image Analysis*, vol. 2652, pp. 847–857, 2003.

[28] N. Morgan and H. Bourlard, "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742–772, 1995.

Chapter 3

Appliance Recognition

Contents

3.1 Introduction

Nowadays, people are very interested in reducing energy consumption. Behind this interest there are several possible motivations, as the prices of energy, political objectives or personal convictions. In this chapter we focus on the electrical consumption of appliances in habitations. More specifically, we address the analysis and modeling of multivariate time series recorded with electrical metering devices. The applications of such modeling are diverse, such as increasing the residents' awareness about their energy consumption.

Smart metering devices are typically able to measure and manage the appliance electrical consumption. These devices are more and more frequently installed in houses, with nowadays a rich selection available on the market [1, 2, 3]. The term "Smart" refers to a range of different applications, e.g. the ability to provide a feedback to the user or to cleverly manage the electrical consumption. *Energy monitoring* provides information to the user to make him aware about the energy consumption. The feedback can be retrieved directly on the smart metering panel or on other devices, as personal computers or mobile phones [4, 5]. The instantaneous consumption and historical data in the form of time series is usually shown. Other types of feedback have been used, as a monitoring energy lamp changing color proportionally to the energy consumption [6]. *Remote control* allows to turn on/off appliances plugged to the smart metering system without necessarily being physically in the proximity. It can be useful for controlling appliances during holidays or prolonged absences from home. *Set scheduling* allows to switch on/off devices depending on a predetermined timing, for instance specific devices can be switched off during night.

In terms of savings, it has been observed that a continuous feedback allows an energy bill reduction ranging from 5% up to 15% [7]. In a realistic scenario, monitoring and feedback systems have to be installed and appliances have to be labeled for being aware to which smart metering devices they are plugged. This problem is actually addressed in this chapter as we will stress that appliances can be automatically recognized by modeling their signatures of electric consumption. In addition, fine-tuned *Automated management* of home equipment allows to anticipate user decisions and improving energy management by analyzing and learning energy consumption patterns.

The Appliance Load Monitoring (ALM) has become a key approach for the consumption understanding and the application of energy saving measures. Two methods have been described in literature:

1. **Non-Intrusive Load Monitoring (NILM)** consists in measuring the electricity consumption using a single metering device, usually called *Smart Meter*. Relying on a single point of measure, it is also called *one-sensor metering*. The qualification of *non-intrusive* means that no extra equipment is installed in the house. All the appliance signatures in the house are aggregated in one single signal and signatures have to be separated for understanding their contribution. This operation is called *disaggregation* of the electricity consumption.

2. **Intrusive Load Monitoring (ILM)** consists in measuring the electricity consumption of one or few appliances using a low-end metering device. The term *intrusive* means that the meter is located inside the house, typically close to the monitored appliances.

As illustrated in Figure 3.1, there is actually a continuum between NILM and the most granular ILM, where only one metering device per equipment is used. We have proposed 3 sub-domains [8]:

ILM I Such systems rely on submeters that are measuring the consumption after the primary utility energy meter. Submeters are typically used to monitor a zone of the house, often corresponding to one of the phase and can be placed at the circuit breaker level. They are represented by the red blocks in Figure 3.2.

ILM II The meters are located at the plug level. The appliances connected to the outlet or multi-outlet are directly monitored. They are represented by the green blocks in Figure 3.2.

ILM III Single appliances are monitored with meters directly embedded in the equipment or in the outlet dedicated to one appliance. Appliances are represented by the blue blocks in Figure 3.2.

Scientific publications go principally in the direction of NILM and ILM III. Papers dealing with NILM outnumber the publications about ILM. Hart and Sultanem [9, 10] started their studies on

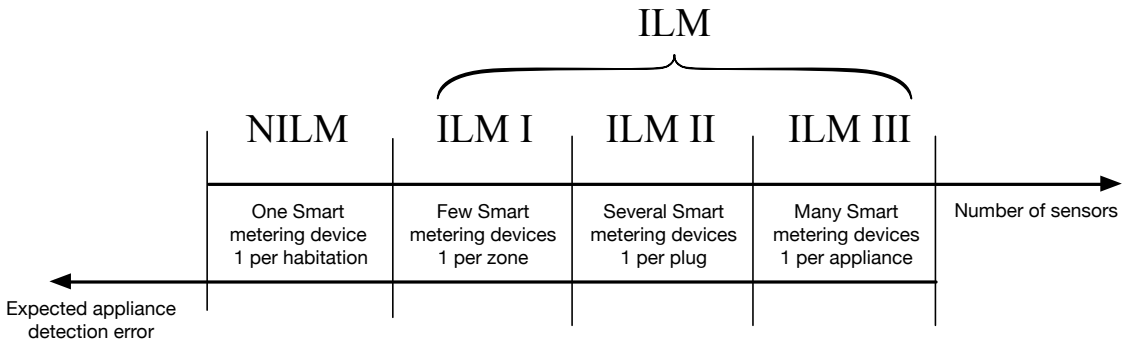


Figure 3.1: Distinction between different categories of ILM and NILM. The different approaches are positioned on the axes of "number of sensors" and "expected appliance detection error", i.e. the more the number of sensors, the finer the acquired information and the more precise the appliance detection.

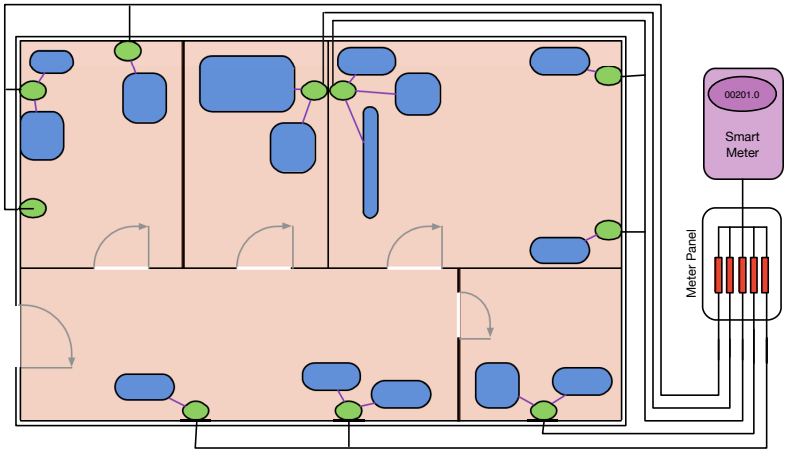


Figure 3.2: An example of a house equipped with several smart metering devices. In blue the single appliances (ILM III), in green the plugs (ILM II), in red the circuit breaker level (ILM I) and in purple the whole home (NILM) are monitored.

NILM in the 80's /90's, making it an elder research topic than ILM. ILM and NILM have different characteristics:

- **Granularity**. With ILM, finer information is acquired as sensors are more numerous by definition. More detailed signatures are also made available and ease their modeling. Typically, low power appliances or stand-by are difficult to identify with NILM [11], while their detection is potentially feasible using ILM. Recently, stand-by power becomes one of the largest source of consumption in residences [12], representing up to 26% of the total energy consumption [13].
- **Installation time**. Given its one-sensor based nature, NILM installation is easier and data acquisition is simpler.
- **Costs**. The installation costs linearly scale with the number of sensors, therefore NILM is the most inexpensive approach. However ILM is typically based on cheap metering devices and, considering the technological progress, their costs are reducing over time.
- **Hard wired appliances**. NILM can be preferred when some appliances are hard-wired, as in case of big appliances (e.g. heating, ventilation, air-conditioning devices). Standard plug-based smart metering devices are typically not used in such cases[1].

[1]However current transducers rolling up the wire exist [14].

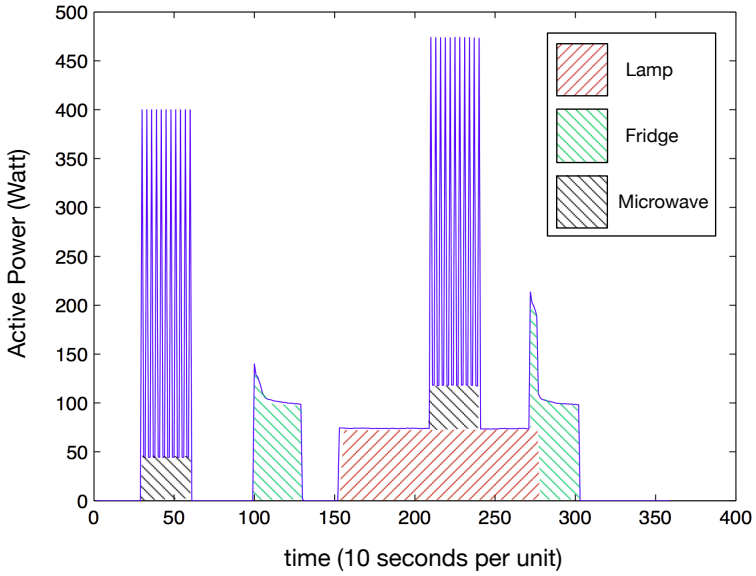


Figure 3.3: An example an aggregated signal. Different appliances (Lamp, Fridge and Microwave) are contributing to the total power consumption $P(t)$.

In conclusion, ILM requires more resources in terms of installation time and costs, however it provides more granular data. In the context of appliance recognition, the accuracy rate is expected to be higher using ILM devices. In addition NILM has some intrinsic drawbacks, i.e. the difficulty to detect low power appliances, appliances in stand-by, appliances with multiple consumption states or those showing continuously variable energy use [11].

In this chapter we focus on the use of ILM devices for the optimization of energy consumption in buildings. In particular, the principal aim is to make the inhabitants aware of the appliance energy consumption. In this way, the inhabitants should be able to better understand their energy consumption and explain their electricity bill.

Part of the Section 3.2 is inspired to our work *A survey on intrusive load monitoring for appliance recognition* [8]. The Section 3.9 is partially inspired to our work *User interaction event detection in the context of appliance monitoring* [15].

3.2 State of the art

Non-Intrusive Load Monitoring

The aim of NILM is mainly to find the contributions of every single electrical appliance to the total power consumption $P(t)$. The total power consumption can be rewritten in terms of the single contributions as:

$$P(t) = \sum_{n=1}^{N_a} P_n(t) \tag{3.1}$$

where N_a is the total number of appliances contributing to $P(t)$. This problem is usually solved by using disaggregation algorithms [16]. In Figure 3.3 an example of an aggregated signal of three appliances, i.e. Lamp, Fridge and Microwave, is illustrated.

The majority of works in literature are based on supervised methods. Two approaches are used for the disaggregation task: (i) optimization or (ii) pattern recognition. The optimization method tries to combine different appliance signatures for obtaining the observed total power consumption $P(t)$. The main disadvantage is that only known appliance signatures are combined, however other unknown signatures could have been acquired. On the other hand, the pattern recognition approach tries to identify typical events of specific appliances inside the total power consumption $P(t)$ [16].

Two approaches are mainly used for computing the appliance features: (i) steady-state and (ii) transient analysis. Several techniques are currently used for the steady-state analysis [17, 18]. The most used is the *Power change* approach, that consists in analyzing the active and eventually the reactive power of the total power consumption. As a drawback, it is hard to separate appliances having similar consumption signatures. Other approaches are based on the analysis of current, voltage and other features in time or frequency. Trajectories in the voltage-current domain or the voltage noise can also be used [16].

Transient approaches search for relevant changes in the power signal and analyze the portion of the signal considered unstable. When devices are turned on or off, some oscillations on the voltage and current signals are produced. These small oscillations are typical of specific appliances and therefore the appliance that has generated them can be identified. The major drawback of this approach is that a high sampling frequency is needed in order to capture the transitions and their effects on signal. Analysis in frequency domain is typically used for discovering transitions and oscillations [19, 20].

NILM can be used for numerous applications:

- **Energy consumption understanding**. NILM provides the total amount of energy consumption, that can directly explain the electricity bill. By using disaggregation algorithms, the contribution of the appliances to the electricity bill can be retrieved [10, 16].
- **Smart energy management**. By observing the trend of the energy consumption during the day, we can understand when high power demanding operations are done. Smart energy management systems can re-schedule these operations when the electricity is cheaper, typically during the night [16, 21].
- **Energy monitoring**. NILM can be used for detecting and diagnosing equipment and component faults through the energy monitoring [22, 23].
- **Human activity recognition**. Users' behaviors can be recognized from the energy consumption of the whole house. In particular, the pattern usage of appliances can be successfully used for this task [24].

As reported in [10], NILM has mainly two setup methods:

1. **MS-NILM**, where MS stands for "Manual Setup". In this setup, appliances are observed and are manually turned on and off for understanding their specific contribution.
2. **AS-NILM**, where AS stands for "Automatic Setup". In this setup only the a priori information about appliances are used without observing their specific contributions, as in the previous case.

MS-NILM requires a certain degree of intrusiveness, because, even if no supplementary hardware is taken inside the home, appliances have to be manually switched on and off. On the other hand, MS-NILM should yield better results.

Intrusive Load Monitoring

Relying on multiple points of measurement, ILM is also referred to as *multi-sensor* metering [25]. Other authors use equivalent denominations such as *Intrusive Appliance Load Monitoring systems* (IALMs) [26, 27], *Hardware-Based Sub-Metering* [28, 29] and *Distributed System / Metering* [30, 31]. We will use *ILM* as generic term.

ILM can be used for numerous applications:

- **Local energy consumption understanding**. The objective is here to provide the household with energy feedback on single appliance. This feedback is typically given directly on the plug or on static or mobile displays [32]. Closely related, some applications are also attempting to predict the electricity consumption of different appliances [33].
- **Global energy consumption understanding**. Related to the first one, another important application is to allow a better understanding of the monthly electricity bill. The principle is to compute the relative contribution of each appliance to the global consumption through an aggregation of the different ILM sensors [34].

- **Appliance monitoring**. The objective is here to detect abnormal electricity consumption, deviation of consumption or faulty devices [35, 36].
- **Evaluation of NILM environments**. ILM can also be employed for evaluating NILM disaggregation performances. In this case, the disaggregation algorithm is applied to the NILM smart meter data and the results are compared to the annotated ground truth provided by ILM sensor data [37, 38].
- **Simulation of NILM environments**. Data acquired from several ILM sensors can also be artificially aggregated to simulate a NILM environment and perform an evaluation of disaggregation algorithms as explained in the previous point [39].
- **Human activity recognition**. ILM has also demonstrated its potential for the indirect detection of human activities through electricity monitoring [40, 41].
- **Appliance localization**. The consumption signature can be used to identify the appliance and the position of ILM sensor provides an indication of its position [42, 43].

Generally speaking, these ILM applications rely on data management, signal processing, statistics and machine learning technologies. An important task for most applications is the matching of incoming unseen signals against previously seen signals. Typically, models are created using historical "training" data and, according to the labels associated to data, one can perform *appliance identification*, *brand identification* and *state identification*. In this section we will provide more details on signal processing and machine learning approaches applied to these domains, pointing to the best-practices in this field.

In a similar way of NILM, we can distinguish two ILM cases for the identification task:

1. **MS-ILM** (Manual Setup ILM) refers to the training of models on data acquired in the environment. A manual intervention of the user is necessary to label "on-site" data according to the appliance. The appliance recognition performances are good as the signals are emitted by the same appliances that were used to train the models. In this case, only temporal variability is impacting the performance of the models.
2. **AS-ILM** (Automatic Setup ILM) do not require the acquisition and labeling of on-site data. The system is a priori trained and may receive signals from "unseen" appliances. Not only temporal variability but also intra-class variability may impact the recognition performance. Intra-class variability is in this case due to difference of brands or models in the same class.

On the one hand, MS-ILM is more expensive than AS-ILM in terms of setup time. On the other hand, MS-ILM is more precise than AS-ILM [29, 44, 45, 46]. Similar conclusions are also reported for NILM in [47].

3.2.1 Data acquisition

A. System architecture

A typical ILM system architecture is illustrated in Figure 3.4. The bottom layer is the network of sensors that may fall in the previously defined categories ILM I (sub-meters), ILM II (plug level) or ILM III (appliance level). Sensors can also behave as actuators with the ability to switch on or off appliances. The next layer is composed of gateways that relay the information from the sensor up to the server. Gateways contain drivers and protocols enabling the communication with the sensors over potentially different networks such as PLC, IP, Enocean or EIB/KNX [41, 48]. The gateway layer is actually optional in the case, for example, of IP based sensors with a direct connection to the server. The server layer is in charge of communicating with the gateways and/or sensors, storing data, processing data and providing the necessary information to render vues in the upper layer. Examples of ILM systems have been described for appliance monitoring [36], appliance identification [49, 44, 30] and consumption prediction [41].

The weakness of such centralized architecture is in the server that is a single point of failure. Other types of systems will probably emerge with more distributed architectures relying on memory and computing capacities embedded in the sensors, typically based on Internet of Things paradigms [50].

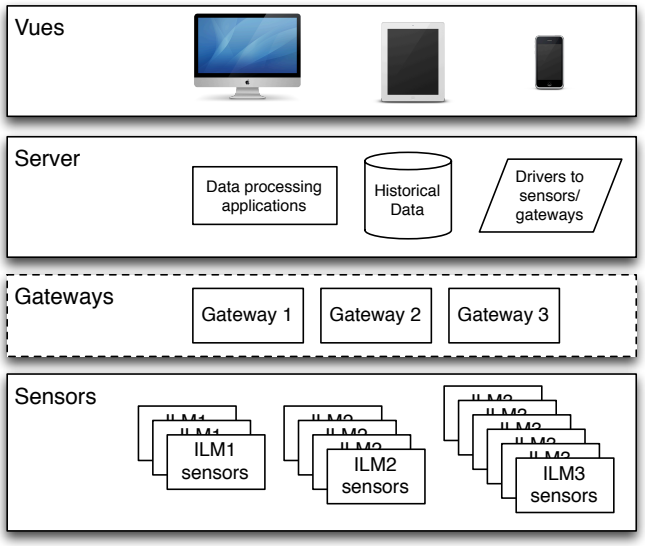


Figure 3.4: ILM system architecture. Sensors, as smart meters at appliance and plug level or sub-meters are communicating to servers through gateways.

B. Appliance characteristics

Limiting ourselves to the residential sector, we can enumerate more than 50 different categories of appliances. Different brands or models from the same category may also show significant differences in terms of electricity consumption. Considering machine learning, the training of models for each category could then become a rather complex task requiring large signature databases. Further to this, appliances can be in different states according to their use or functioning. Appliances can be grouped in four categories as explained in [10] and updated in [51, 16]:

- **Type I (on/off)** has typically only two states either on or off, such as light bulbs or toasters.
- **Type II (finite states)** has a finite number of different modes of energy consumption, such as microwaves, fridges and tv sets.
- **Type III (continuously variable)** varies the electrical consumption as a function of time, typically for charging batteries such as laptops or phones.
- **Type IV (permanently on)** has a constant consumption, as alarms or landline phones.

In Table 3.1, we illustrate the diversity of appliance categories used for identification tasks in different research work. For sake of simplicity, the categories with at least five occurrences across research works are considered. This table clearly shows a large variability, making difficult to compare the different research works in terms of performances.

Depending on the target application, the sampling frequency of the electricity consumption may vary. In some works, a rather high sampling frequency from 1 kHz up to almost 100 kHz is used [44, 58]. In this case, the identification results are more accurate, given that high frequencies allow to capture finer state transitions and eventually to separate brands belonging to the same category. As a drawback, the equipment of such systems is more costly [59]. Other works have shown the possibility to use cheaper meters with limited sampling frequencies, typically less than 10 Hz [56, 46, 29].

3.2.2 Available databases

The number of NILM databases that are publicly available is larger than ILM. Kolter et Johnson [60] present the Reference Energy Disaggregation Data-Set (REDD). This data-set contains several types

	[52]	[31]	[53]	[54]	[55]	[56]	[44]	[29]	[57]	[49]	[58]	[30]
Fridge	x	x	x	x	x	x	x	x	x		x	
Microwave	x		x	x		x	x	x				
Dishwasher	x		x					x	x			
Wash machine			x				x	x	x			
Coffee mach.	x		x		x	x		x	x			
Computer	x		x	x	x	x	x		x	x		
Laptop		x	x	x	x	x			x		x	x
Printer	x		x	x		x			x			
Monitor		x	x	x					x		x	
Television		x	x	x		x	x	x	x	x	x	
Lamp		x	x	x		x	x	x	x	x	x	x
Vacuum cleaner			x	x			x		x	x		
Fan				x			x		x		x	x
Iron			x				x	x		x		
Hair dryer				x			x		x	x		
Phone charger				x	x	x			x	x	x	
Shaver						x			x	x		
Others		x	x	x			x	x	x	x		x

Table 3.1: Appliance categories used in different research works. For sake of simplicity, we do not report the differences between the following appliances: LCD monitor and CRT monitor (indicated as *monitor*), LCD television and CRT television (indicated as *television*), laser printer and ink-jet printer (indicated as *printer*), fluorescent lamp and halogen lamp (indicated as *lamp*). We merge also the categories *fridge* and *freezer* that in some works are explicitly separated and in others are not.

of measurement: the whole home electricity signal acquired at sampling frequency of 15 kHz, 24 individual circuits in the house with information about the appliances (measured at 0.5 Hz) and finally 20 plug-level monitors in the house recorded at 1 Hz. They present an initial benchmark using Factorial Hidden Markov Models. Makonin et al. [61] describe the Almanac of Minutely Power dataset (AMPds) which contains the measurement of 11 features by using 21 sub-meters, positioned at breaker level on the power panel. The sensors acquire a sample every minute for one year. Kyle et al. [62] propose the Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) which includes the measurements of a whole-house consumption in terms of voltage and current by using a sampling frequency of 12 kHz. They provide the ground truth for every state transition of the appliances. Moreover, they present an initial benchmark using a modified generalized likelihood ratio detector. Monacchi et al. [63] create an energy consumption data-set of 9 households in Italy and Austria, called GREEND, that provides the consumption data of selected devices. They also provide different scenarios to use the data-set, like disaggregation, occupancy detection and appliance usage modeling. Kelly et Knottenbelt [64] present the UK Domestic Appliance-Level Electricity (UK-DALE) dataset. It contains the electricity consumption of the whole house at a sampling frequency of 16 kHz and of 1/6 Hz for individual appliances. They record four houses for several days.

Other data-sets contain information coming from highly equipped houses having different kind of sensors. Such data can be used for appliance recognition, cost optimization, prediction and more. Barker et al. [65] create the Smart* Home Data-Set that contains data from three houses. They provide the electricity at the main panel, electricity at outlets, renewable generation, wall switch events, thermostat events, motion events, door events and weather station data. Batra et al. [66] release their collection of data, called Indian data for Ambient and Electricity Sensing (iAWE), which consists in the acquisition of as much data as possible from a smart environment. Electricity, water, ambient and other parameters are constantly monitored. Regarding electricity, the consumption is monitored at the meter level, circuit level and appliance level. Voltage, current, frequency, power (active and apparent), energy and phase are monitored for 10 appliances with a sampling rate of 1 Hz.

We observe an increasing interest of the scientific community for ILM applications. Many research works report on the use of data recorded through acquisition campaigns using commercial sensors or specific hardware such as in [67, 68, 35]. Publicly available databases can be used, allowing fair comparison of systems. Dedicated to appliance recognition, the *Tracebase*[2] database contains more than a thousand of electrical appliance signatures, recorded from 122 appliances spread into 31 categories and acquired with a sampling frequency between 1 and 10 Hz. [53]. Gao et al. [69] introduce

[2] www.tracebase.org

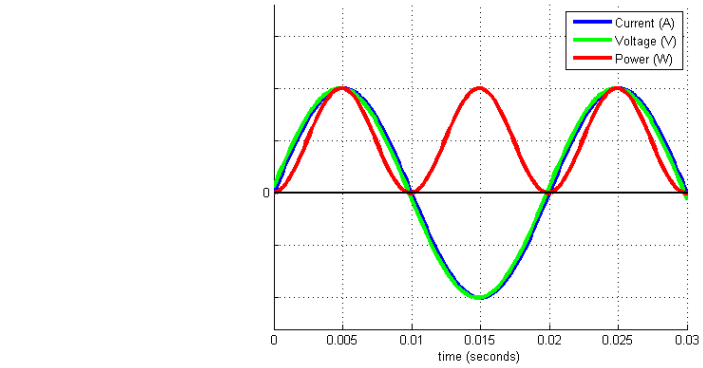
the Plug-Level Appliance Identification Data-set (PLAID) that contains short term measurements of several appliances. The dataset contains more than one thousand measurements from 235 appliances spread into 11 appliance classes. The voltage and current consumed by the appliances are measured with a sampling frequency of 30 kHz. The data-set is public and crowd-sourced. Other databases have been made available but are rather dedicated to statistical analysis [70, 71].

3.2.3 Feature Extraction

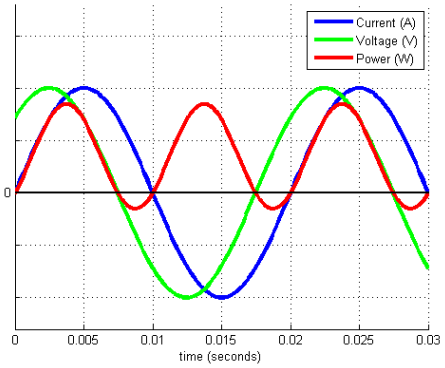
Electricity in Europe and the majority of Africa and Asia is delivered as alternating current (AC) at a frequency of 50 Hz. However, it can be different depending on the country, for example in North and Central America the frequency is 60 Hz. This frequency is usually called **utility frequency** or **power line frequency**. Considering a power line frequency of 50 Hz, alternating **current** and **voltage** waves transmitted in the power lines have a period of 1/50 of a second. They alternate positive to negative cycles, meaning that the flow of electric charge periodically reverses direction. The difference of **phase** between the current and the voltage (φ) depends on the circuit characteristic, represented by three main circuit components: the resistor, the capacitor and the inductor. When the current and voltage are synchronous, φ is equal to zero and appliances have a resistive behavior. This is the case of appliances with heating components, as kettles, dishwashers and coffee machines. Many appliances have a hybrid behavior. If the voltage is delayed from the current, then φ is greater than zero and appliances, as fluorescent tubes, refrigerators and fans, have an inductive behavior. When the opposite occurs the φ, is smaller than zero and appliances have a capacitive behavior, as CRT TVs. A representation of φ is shown in Figure 3.5.

A useful way of representing the current and the voltage is the **RMS**[3] **current** (I_{rms}) and **RMS voltage** (V_{rms}), that are the values of current and voltage produced by direct current having the same power. Such a representation is useful when measuring the alternative current and voltage with

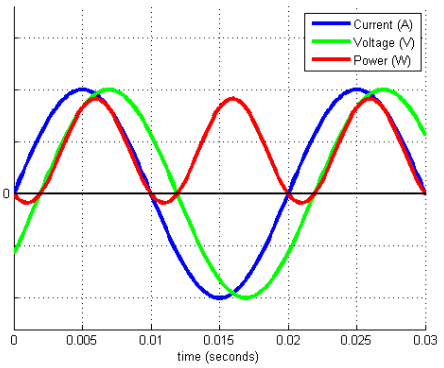
[3]RMS stands for root mean square



(a) Resistive behavior

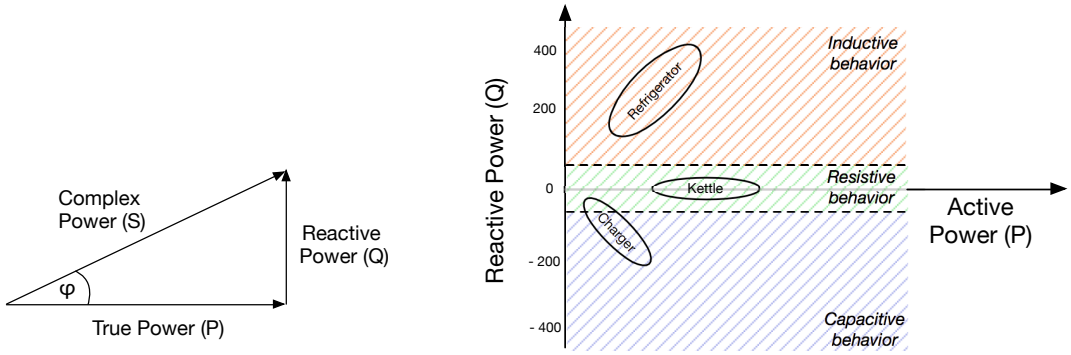


(b) Inductive behavior



(c) Capacitive behavior

Figure 3.5: Different behaviors of the voltage and current



(a) Different types of powers

(b) Distribution of some appliance in the P-Q plane

Figure 3.6: The relation between active and reactive power.

a low sampling metering system. The voltage can be seen as the force necessary to move the electrons, while the current can be seen as a rate of flow of the charge of electrons per second through a material depending on the voltage appliance. By multiplying the force by a rate we obtain an energy expanded over time, namely the **power** [72]. The electrical power is measured in Watt (W) and it can be computed by the following Formula:

$$P = V_{rms}\ I_{rms}\ cos(\varphi) \tag{3.2}$$

For the appliance having a resistive behavior, the power is always positive, meaning that the appliance only consumes the power. Otherwise, the voltage and the current have periodically different signs. This generates a negative power, meaning that the power is partially stored and later returned to the source.

There are different types of power:

- **True** (or **Active**, or **Real**) **Power** (P) is the power performing the work. It is measured in Watt (W) and is the result of resistive components.
- **Reactive Power** (Q) is the power not performing the work. It is measured in reactive Volt-Ampere (VAR) and is the result of capacitive and inductive components.
- **Complex Power** (S) is the vector sum of the active and reactive power, as shown in Figure 3.6a. It is measured in Volt-Ampere (VA).
- **Apparent Power** ($\mid S \mid$) is the magnitude of the Complex Power. It is measured in VA.

Some appliances can be distinguished in the P-Q plane depending on their resistive, capacitive and inductive characteristics [73]. As shown in Figure 3.6b, appliances distribute in different zones of the P-Q plane: resistive appliances, as kettles, do not have any reactive power (the voltage and the current are synchronous, therefore φ is equal to zero). For this reason, resistive appliances distribute on the x-axis of the P-Q plane. Capacitive appliances, as chargers, generate signatures where the current is delayed from the voltage and therefore φ is negative. These appliances form clusters in the bottom part of the P-Q plane. Inductive appliances, as fridges, generate signatures where the voltage is delayed from the current and therefore φ is positive. These appliances form clusters in the upper part of the P-Q plane.

In literature these basic features are currently used for the appliance recognition task. The majority of papers uses power-based features given their popularity among the commercial smart metering devices. From these features, others can be computed. The voltage-current (V-I) trajectories have been proposed for characterizing the appliances [54]. The peak values of the current are also used, for example in [49]. Information about the current form, as the crest factor [58], the form factor or their combination have been used in [49]. The inrush current compared to the steady state current appears in [31].

Frequency features are used in the context of appliance recognition when the acquisition frequency is medium to high (from 1 kHz up to 100 kHz). Discrete Fourier Transformation (DFT) or Fast

Fourier Transformation (FFT) are typically applied. DFT is reported to be less efficient when the sampling frequency is low [52]. FFT is used in [53] to determine the highest frequency in the signature and the energy contained in the band of 5%, 15% and 25% of the highest frequency. The impact of FTT-based features has been evaluated showing differences in performances of the system when injecting other features derived from the current in time [31]. Using FFT on signals with a sampling frequency of 20 kHz, appliance harmonic content is unique and even devices of the same category can be distinguished [74]. Other authors reported detailed analysis of harmonics using high sampling frequency [58].

Pre-processing operations such as normalization and phase shifting are sometimes applied [57]. In some cases, a reduction of the feature space is also applied, typically using Principal Component Analysis (PCA) [44]. In our works, we propose to use features based on the signal evolution, as the first and second derivative [56, 46]. As discussed later, such features are bringing useful information in the classification task. Other features are based on counting the number of occurrences of events in a period of time, such as the number of transitions between power intervals in [29], the number of edges in [52] and the number of threshold crossings in [53]. Other features, as the maximum, minimum or average of power in a certain interval of time have been used [29, 53].

3.2.4 Machine Learning

Most of the machine learning approaches used in the context of appliance identification are based on supervised techniques. Paradiso et al. [29] are proposing the use of multilayer perceptrons trained with back-propagation. They use low sampling frequency data from which power-based features are extracted, such as the number of samples and transitions between power intervals. Tests are reported for MS-ILM and AS-ILM systems. Accuracy of 95.3% is reported for MS-ILM when using 8 appliances. Kato et al. [44] propose to use one-class Support Vector Machines (SVM) with Gaussian kernels. The features are extracted by voltage and current wave-shape using a sampling frequency of 6 kHz. In the MS-ILM case, they use 16 appliances achieving 99.9% of accuracy rate. In the AS-ILM case they use 25 appliances achieving an identification rate of 95.8%. Zufferey et al. [55] compare k-Nearest Neighbors (K-NN) and GMMs using 6 appliance categories. The electric measurements are sampled at 10^{-1} Hz using low-end Plogg meters and the features are power-based. In their best case, they report an accuracy rate of 85% when using K-NN. Reinhardt et al. present in several works [53, 31, 75] the comparison of various algorithms as Bagging, Bayesian Network, J48, JRip, LogitBoost, Naive Bayes, Random Committee, Random Forest and Random Tree for the classification of appliance categories using a MS-ILM scenario. In a first work, they use data from the *Tracebase* database recorded at low frequency. The best reported performance is up to 95.5% accuracy using Random Committee. In another work, the same classification algorithms are used on higher frequency data (1.6 kHz) from which current-based and harmonics features are extracted. Better performances are reported up to 100% for Bayesian network. Englert et al. [58] present a hardware and software system treating high frequency data (96 kHz). Temporal and frequency features are computed from the voltage and current values. Different tasks are evaluated including category recognition, equipment identification and operation mode recognition. For category recognition, 99.8% accuracy is reported using Random Forest classifiers on 14 categories. In the work of Adeel Abbas Zaidi and Palensky [52], Dynamic Time Warping (DTW) and HMMs are proposed. The signatures are sampled at 10^{-1} Hz and several features are extracted and benchmarked. HMMs are reported to perform better than DTW using a 6 category task. Saitoh et at. [57] report on the use of K-NN for the identification of 35 appliances in a MS-ILM scenario. The signatures are sampled at 4.4 kHz from which current-based features are extracted. Accuracy is reported for category recognition, appliance recognition and state recognition, with, respectively 85.5%, 80.5% and 76.3%. Fitta et al. [76] propose to recognize the switch-on and off events from appliance signatures acquired at 6.4 kHz. They used K-NN systems trained on extracted features such as the real power and current harmonic. Performance on this 2-class task of 87.5% and 90.6% correct identification is reported for, respectively, switch-on and switch-off events. Other works reporting rather few algorithmic details on ILM tasks can be found in [30, 49].

Rather few unsupervised techniques are reported for ILM tasks. Lam et al. [77, 54] propose the use of Dendrograms for appliance representation and clustering. Shape features of trajectories in the V-I space are used as features. The Dendrograms allow to cluster 30 appliances into 13 groups showing similar operating characteristics.

We observe that machine learning algorithms are well spread in papers without a preferred technique. The appliances and features choice have consequences on the pattern separability in the feature space and therefore it influences the selection of machine learning algorithms.

3.2.5 Discussion

We observe an increasing number of papers published over the last 4 years about ILM tasks, showing an increasing interest of the scientific community. About 50 papers related to load monitoring and more specifically to ILM allow us to draw the following conclusions:

1. **Data availability**. ILM currently lacks of large databases that are publicly available, especially if compared to NILM. Comparison between scientific works is currently made difficult as many teams work on their own private data sets using specific protocols. The recent appearance of new public available databases will potentially foster the scientific community towards comparable evaluations.

2. **Appliance sets**. The type of appliances and their occurrences in the data sets are probably influencing the reported performances in a significant manner. In general, appliances having similar resistive, capacitive and inductive characteristics are more difficult to separate in the feature space. This fact leads to two consequences. First, one should analyze the types of appliances presented in scientific papers to fairly interpret the reported performances. However, such information is sometimes not available in papers. Second, current reported results are probably optimistic as larger data sets with more categories will show lower performances.

3. **Sampling frequency**. In general, we observe that higher sampling frequency of signatures give better classification results. More detailed signal information is indeed available, allowing to use frequency-based features that are especially powerful for appliance identification inside the same category. However, medium-high frequency metering equipments are more expensive than low frequency sampling meters.

4. **Machine learning**. There is currently no "leading" or preferred machine learning technique. This is probably due to the variability of the sets of appliance categories, and due to the different types of measurements, sampling frequencies and feature selections.

3.3 ACS-F Database

3.3.1 Introduction

In the last years, we observe an increasing interest of the scientific community for ILM applications. Unfortunately, in literature there are few available ILM databases, therefore researchers have to use them or record their own set of data. In order to address the need for data, we have created the *Appliance Consumption signature - Fribourg* (ACS-F) database, available in two versions: ACS-F1 and the ACS-F2 [45, 78]. We have hosted the databases on `http://www.wattict.com` site. People interested in the databases have to fill a licensing form and accept the use conditions.

We grant free access to both versions of the database:

- ACS-F1 is the first version of the *ACS-F* database [45]. The database contains 200 electric signatures acquired from 100 appliances. The signatures are uniformly spread in 10 categories.

- ACS-F2 is the second version of the *ACS-F* database [78]. It can be considered as an extension of the ACS-F1. The ACS-F2 version contains 450 signatures acquired from 225 appliances uniformly spread into 15 categories.

In Figure 3.7 the differences between the two databases is represented.

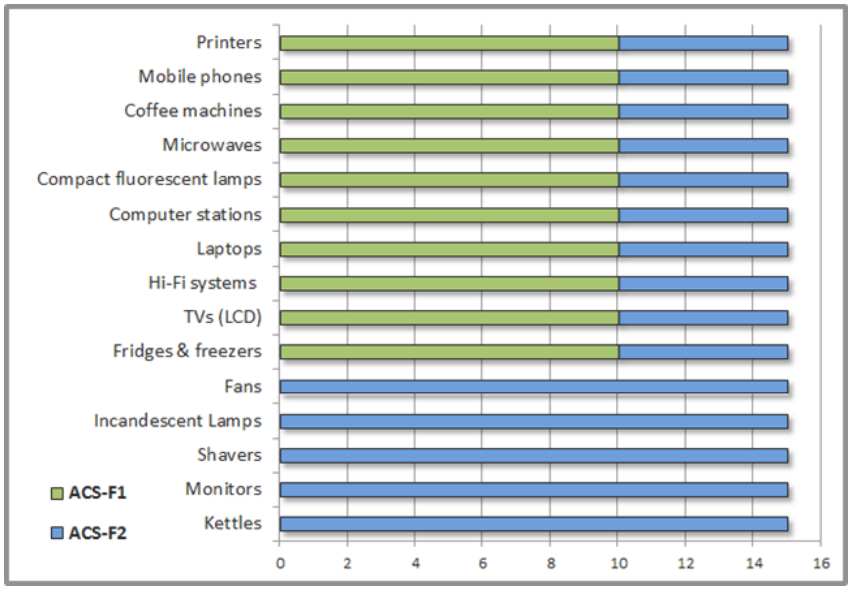


Figure 3.7: Current versions of the *ACS-F* database: ACS-F1 (light green) and ACS-F2 (blue).

3.3.2 Data acquisition protocol

As explained in Section 3.2, the databases reported in literature show different conditions of recording, such as the types of appliance, sampling frequency, features, etc. The reported results are also very different. In our opinion it is very useful to clarify as much as possible the conditions of recordings that we resume hereunder:

Sampling Frequency
In our database we use a low sampling frequency, namely 10^{-1} Hz. As previously stated, higher sampling frequency yields better classification results because signatures are represented through finer information. However, low frequency metering equipment is cheaper and consumes less energy.

Features
We record six different features: *active power*, *reactive power*, *RMS voltage*, *RMS current*, *phase angle* and *frequency of the network*.

Acquisition duration
Every appliance in the database has been recorded 2 times for 1 hour. In the reminder of the text, we refer to these 2 recordings of 1 hour as *session 1* and *session 2*. During the acquisition, the appliances have been normally used. By normally, we mean that we have somehow respected the different modes and duration of use in typical situations (see below).

Number of categories
Depending on the version of the database, we have a different number of categories:

- 10 for the ACS-F1: Mobile phone Chargers, Coffee machine, Computer workstation (including monitor), Fridge and freezer, Hi-Fi system (CD players), Lamp (CFL), Laptop (via chargers), Microwave oven, Printer, and Television (LCD or LED).
- 15 for ACS-F2: Mobile phone, Coffee machine, Computer station (including monitor), Laptop, Fridge and freezer, Hi-Fi system (CD players), Lamp CFL, Incandescent lamp, Microwave oven, Printer, Television (LCD or LED), Monitor, Kettle, Fan and Shaver

Number of appliances per category
Also the number of appliances per category depends from the version of the database: (i) 10 appliances per category for the ACS-F1 and (ii) 15 appliances per category for the ACS-F2. All

Figure 3.8: Plogg device.

appliances inside the same category have different brand and/or model, therefore there are no repetitions inside the database.

Smart metering device
We use a Plogg device based on Zigbee technology (produced in 2008 by Energy Optimizers Limited). The device is shown in Figure 3.8.

For every appliance category we have established a precise acquisition method:

1. *Printer* must be in operating mode, standby and turned off. The number of times the printer is used and the number of papers can vary.
2. *Mobile phone* (via charger) can have whatever battery level, from completely uncharged to fully charged. Mobile phones can be used or unplugged during recording.
3. *Coffee machine* must be in operating mode, standby and turned off. When used, the size of the cup and the type of coffee can vary.
4. *Microwave* must be in operating mode, standby and turned off. User must set different modes of the oven with different power settings for a variable time. If the microwave has other functionalities such as grill, they are also used.
5. *Lamp (fluorescent)* must be in operating mode, standby and turned off.
6. *Computer workstation* (including monitor) must be in operating mode, standby and turned off. Some activities having different working loads, e.g. surfing on the web or watching films, are systematically performed by the user. Changes of the brightness of the monitor are optionally performed.
7. *Laptop* (via charger) must be in operating mode, standby and turned off. As the computer workstation, some activities having different working loads must be performed by the user. Changes of the brightness of the monitor can be as well effectuated.
8. *Hi-Fi system (CD player)* must be in operating mode, standby and turned off. User must play CD, listening radio (if present) and increasing or decreasing the volume.
9. *Television* (LCD or LED) must be in operating mode, standby and turned off. User must change the channel and increasing or decreasing the volume.

10. *Fridge and freezer* must never be turned off, i.e unplugged. During the acquisition the fridge must be used by inserting or removing food.

11. *Fan (mechanical)* must be in operating mode, standby and turned off. All the possible velocities of the fan must be tested for a variable duration of time.

12. *Lamp (incandescent)* must be turned on and off during the acquisition.

13. *Shaver* (via charger) must be in operating mode, stand by and turned off. All the possible velocities of rotating or oscillating blades have to be tested.

14. *Monitor* must be in operating mode, stand by and turned off. As for computer stations and laptops, different activities must be performed by the users, as surfing on the web or watching films. Changes of the brightness can optionally be operated.

15. *Kettle* must be turned on and off during the acquisition. Different random levels of water are boiled during the acquisition.

The ACS-F1 database includes the appliance categories from point 1 to 10, while the ACS-F2 includes all the categories.

3.3.3 Data format

Our data is available in two formats: .XML and .mat.

The XML data structure contains meta-data information, stored in the header, and raw observations, stored in the body. The meta-data contains information about the version of the database ("1" for ACS-F1 and "2" for ACS-F2), the session ("1" or "2"), the author of the acquisition, the date, the acquisition place, the sensor device and finally the electrical parameters acquired. The body contains the values of the raw observations and their time-stamps. Hereafter an example of an XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<signalData>
  <acquisitionContext database="ACS-F1"  session="1" version="1" />
  <validation status="not" id="antonio" date="2012-01-23" />
  <acquisitionPlace name="myRoom" type="kitchen">
    <feature name="buildingName" value="antonio house" />
    <feature name="buildingType" value="residential" />
    <feature name="energyClass" value="?" />
    <feature name="constructionYear" value="?" />
    <feature name="rooms" value="?" />
    <feature name="surface" value="?" />
    <feature name="address" value="ITA-50132 Florence" />
  </acquisitionPlace>
  <targetDevice type="Fridge" brand="Miele" model="F7261S-1" energyClass
      =""
                        comment="fridge in my kitchen" />
  <acquisitionDevice type="electricity_socket" brand="PLOGG" model="PLG-
      ZGB-CH"
                samplingFrequency="0.1" comment="Sampling frequency is
                    expressed in Hz.">
    <channel name="time" type="date" precision="1" units="s" />
    <channel name="freq" type="float" precision="0.1" units="Hz" />
    <channel name="phAngle" type="integer" precision="1" units="" />
    <channel name="reacPower" type="float" precision="0.001" units="var"
       />
    <channel name="rmsCur" type="float" precision="0.001" units="A" />
    <channel name="rmsVolt" type="float" precision="0.001" units="V" />
    <channel name="power" type="float" precision="0.001" units="W" />
  </acquisitionDevice>
```

```
  <signalCurve>
    <signalPoint time="2010-12-10 16:49:31" freq="49.9" phAngle="29"
        reacPower="47.101" rmsCur="0.429" rmsVolt="224.709" power="82.298"
         />
    <signalPoint time="2010-12-10 16:49:41" freq="49.9" phAngle="29"
        reacPower="46.687" rmsCur="0.425" rmsVolt="224.15" power="81.159"
        />
    <signalPoint time="2010-12-10 16:49:51" freq="49.9" phAngle="29"
        reacPower="46.894" rmsCur="0.423" rmsVolt="224.783" power="80.745"
         />
    <signalPoint time="2010-12-10 16:50:01" freq="49.9" phAngle="30"
        reacPower="46.998" rmsCur="0.421" rmsVolt="225.105" power="80.331"
         />
    <signalPoint time="2010-12-10 16:50:11" freq="49.9" phAngle="30"
        reacPower="47.101" rmsCur="0.414" rmsVolt="225.605" power="78.571"
         />
       ...
    <signalPoint time="2010-12-10 17:48:31" freq="50.0" phAngle="0"
        reacPower="0.0" rmsCur="0.0" rmsVolt="227.259" power="0.0" />
    <signalPoint time="2010-12-10 17:48:41" freq="50.0" phAngle="0"
        reacPower="0.0" rmsCur="0.0" rmsVolt="227.902" power="0.0" />
    <signalPoint time="2010-12-10 17:48:51" freq="50.0" phAngle="0"
        reacPower="0.0" rmsCur="0.0" rmsVolt="227.647" power="0.0" />
    <signalPoint time="2010-12-10 17:49:01" freq="50.0" phAngle="0"
        reacPower="0.0" rmsCur="0.0" rmsVolt="228.166" power="0.0" />
    <signalPoint time="2010-12-10 17:49:11" freq="50.0" phAngle="0"
        reacPower="0.0" rmsCur="0.0" rmsVolt="227.652" power="0.0" />
  </signalCurve>
</signalData>
```

The MAT data structure contains only the raw observations with the advantage to be easily readable by MATLAB and Octave softwares. Hereafter and example of a MAT file:

```
    # name: phAngle
    # type: matrix
    # rows: 1
    # columns: 359
       29 29 29 30 30  ...   0 0 0 0 0
    # name: freq
    # type: matrix
    # rows: 1
    # columns: 359
       49.9 49.9 49.9 49.9 49.9   ...   50.0 50.0 50.0 50.0 50.0
    # name: reacPower
    # type: matrix
    # rows: 1
    # columns: 359
       47.101 46.687 46.894 46.998 47.101   ...   0.0 0.0 0.0 0.0 0.0 0.0
    # name: power
    # type: matrix
    # rows: 1
    # columns: 359
       82.298 81.159 80.745 80.331 78.571   ...   0.0 0.0 0.0 0.0 0.0 0.0
    # name: rmsVolt
    # type: matrix
    # rows: 1
    # columns: 359
```

```
    224.709 224.15 224.783 225.105 225.605   ...   227.259 227.902
       227.647 228.166 227.652
# name: rmsCur
# type: matrix
# rows: 1
# columns: 359
    0.429 0.425 0.423 0.421 0.414   ...   0.0 0.0 0.0 0.0 0.0
```

In addition we provide the images of the different features in two formats: SVG and ESP. SVG (Scalable Vector Graphics) is based on XML and usually can be opened by all major web browsers. EPS (Encapsulated PostScrips) is a PostScript document that contains the description of the image. For every signature there are the images of all the features in both formats.
The nomenclature of the files has been chosen depending on several parameters: (i) the initial letters of the user performing the recording, (ii) the appliance category, (iii) the brand of the appliance, (iv) the model of the appliance and (v) the session number. Usually all the parameters are present, however it could happen that the information about the brand and/or model is not available for some appliances. In such cases, the corresponding fields have been omitted. An example of a standard nomenclature is shown in Figure 3.9. In Figure 3.10 an example of signature for a Fridge signature is illustrated.

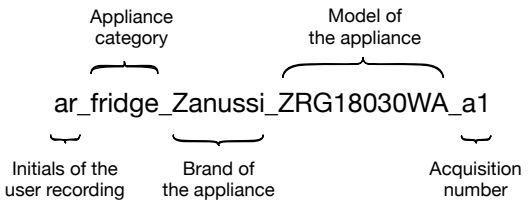


Figure 3.9: Example of the file nomenclature.

3.3.4 Data visualization

In many cases appliances can be successfully discriminated considering simple electrical features, as real power (P) and reactive power (Q) [79]. In the P-Q plane appliances tend to form clusters depending on their circuital characteristics, as explained in Section 3.2.3. In Figure 3.11 all the samples contained in the ACS-F2 database have been plotted on the P-Q plane using different scales on the axes.

By observing the data on the P-Q plane, we verify that several points are concentrated near the (0,0) point. In fact, these points are principally related to stand-by states, off states and the charged states of battery devices. These states can mostly be considered as "stables", i.e. only small variations are verified. Consequently, we add to the signatures the information about their evolution in time, namely the velocity and acceleration coefficients, as will be further explained.

3.4 Pre-processing operations

3.4.1 Dynamic coefficients

In our experiments, we opt to include information about the signal dynamics through the computation of *delta* (or *velocity*) and *delta-delta* (or *acceleration*) coefficients. As discussed in Section 3.2.1, such coefficients have the potential to characterize equipments showing a continuously variable consumption profile (Type III). These coefficients are also able to capture information about transitions between modes of operation. They are commonly used in many fields, as in speech recognition [80]. We

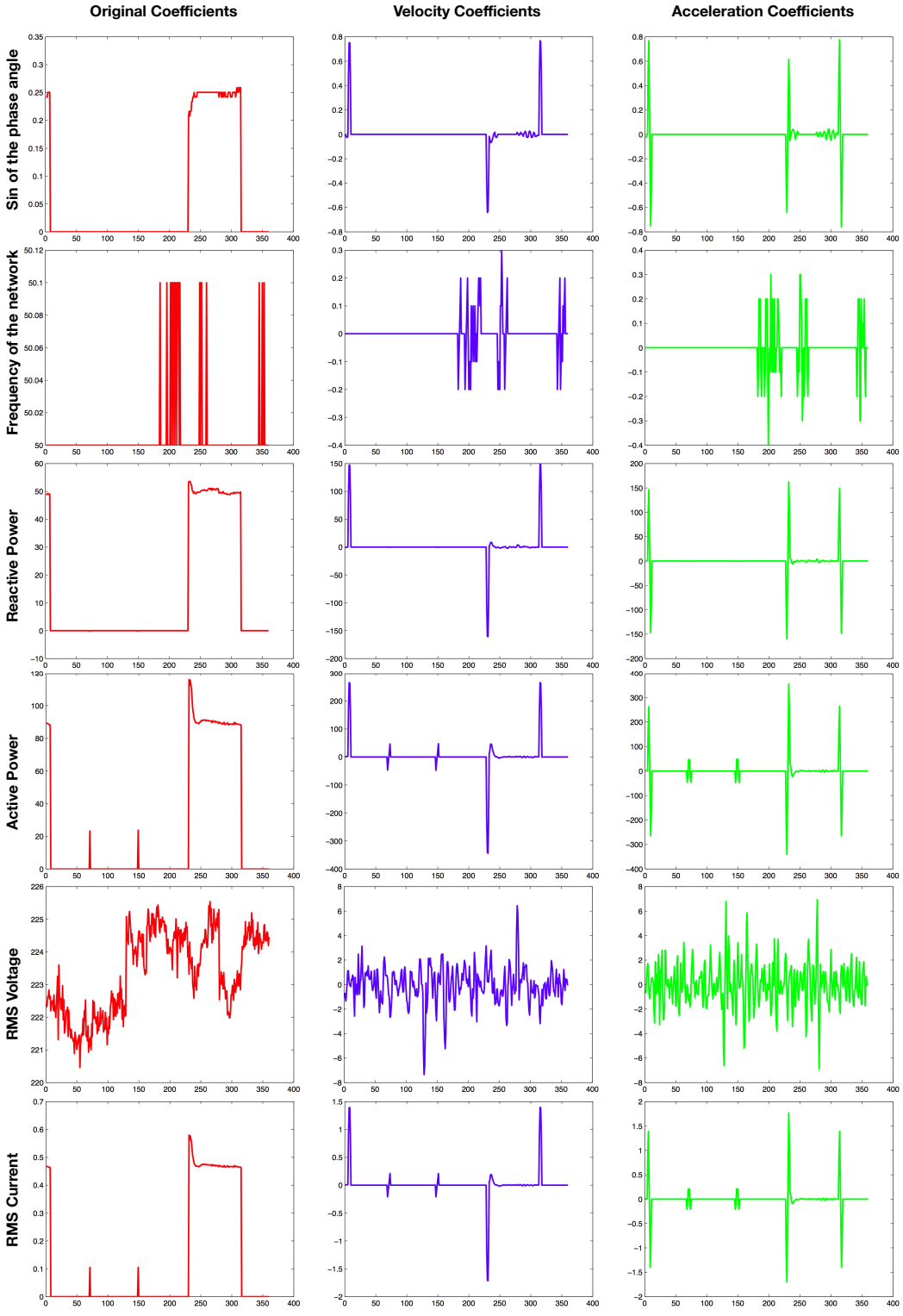
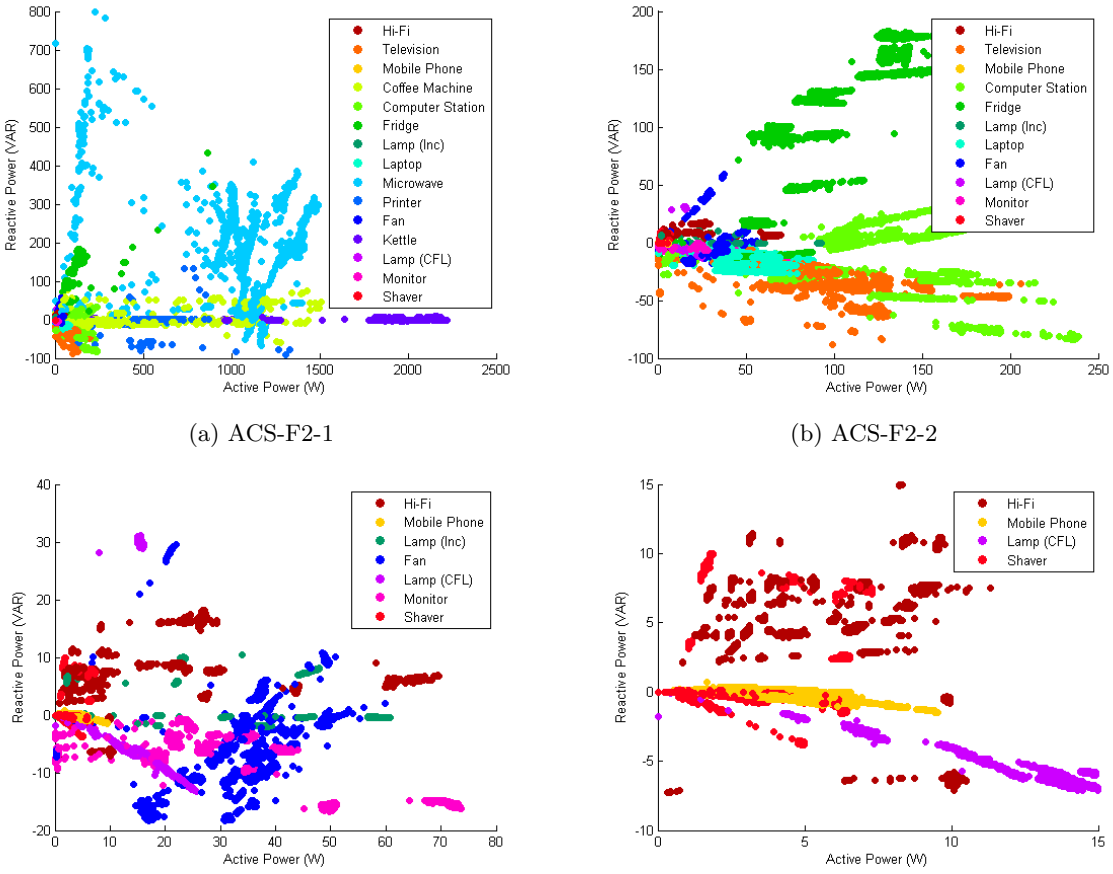


Figure 3.10: Example of all the features of a Fridge signature in the ACS-F2 database. The columns represent in the order the original features, the velocity coefficients and the acceleration coefficients. The velocity and acceleration coefficients are not present in the ACS-F database. However, given their extensive use in the classification task, they have been computed and added to the Figure.



(a) ACS-F2-1

(b) ACS-F2-2

(c) ACS-F2-3

(d) ACS-F2-4

Figure 3.11: Different scales on the axis on the ACS-F2 database. In the first part (a) all the appliances in the original scale are represented. In (b), (c) and (d) we gradually reduce the scale of the axes by selecting only the appliances having all their points in the plot area.

compute the *delta* coefficients as follows:

$$\Delta o_n = \sum_{w=-W}^{W} w o_{n-w} \tag{3.3}$$

where W is the window length. The acceleration coefficients can be computed from the delta coefficients:

$$\Delta\Delta o_n = \Delta o_{n+1} - \Delta o_{n-1} \tag{3.4}$$

The dynamic coefficients need a window of time for their computation, as illustrated in Figure 3.12. The number of samples used for computing the velocity coefficients is $W_d = 2W + 1$, while that for the acceleration coefficients is $W_{dd} = 2W + 3$.

Considering a multivariate time series having D dimensions, the n-th observation vector can be extended as follows:

$$o_n = \{o_{1n}, \ldots, o_{Dn}, \Delta o_{1n}, \ldots, \Delta o_{Dn}, \Delta\Delta o_{1n}, \ldots, \Delta\Delta o_{Dn}\} \tag{3.5}$$

The impact of including the dynamic coefficients on the accuracy rates will be later investigated. In the second column of Figure 3.10 we report the velocity coefficients and in the third column the acceleration coefficients.

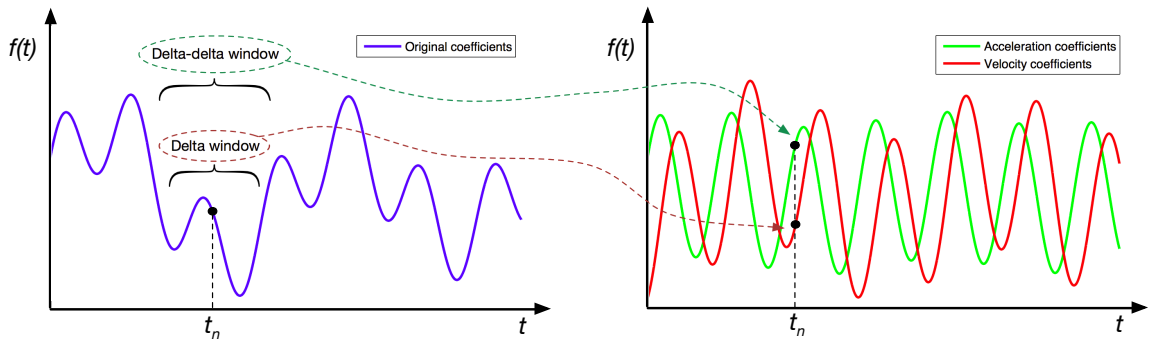


Figure 3.12: Computation of the dynamic coefficients for a synthetic signal. A given n-th sample value is extended through two new values computed on the *delta* and *delta-delta* windows.

3.4.2 Normalization

A typical operation applied to the features is the z-normalization:

$$x_{ki} = \frac{o_{ki} - \mu_k}{\sigma_k} \tag{3.6}$$

where μ_k and σ_k are respectively the mean and variance computed for every feature using all the examples in the training set. After this operation, the mean is equal to zero and the variance is equal to one. The z-normalization is applied on both the original and the dynamic coefficients. For some machine learning algorithms as the K-NN, the normalization is usually performed because distances are influenced by the measurement units and otherwise features would have different weights depending on their scale.

3.4.3 Phase shifting

The electric phase φ is described by an angle between 0 and 360 degrees. Because of the cyclic nature of the phase, some appliances can have consecutive samples in time that shift from 0 to 360 degrees and vice versa. For avoiding this jump in the phase value we compute the sinus of φ in radiant. The problem of bijection is excluded by the positivity of the cosine of the phase in radiant of all the samples. For sake of simplicity hereunder we will use the term "phase angle" for indicating the sinus of φ in radiant.

3.5 Evaluation protocols

To evaluate our own algorithms and to compare our results with those of other researchers, we have created different evaluation protocols. Doing so, researchers should be capable to repeat the same experiences as ours and to compare their own machine learning algorithms. We create two categories of protocol: (i) fixed time length and (ii) variable time length. The first allows to use the whole duration of the signatures while the second analyzes portions of the original signatures by using an analysis window. In these two categories, we also have proposed two sub-protocols corresponding to MS-ILM and AS-ILM usage presented in Section 3.2. More details are provided in the next sections.

3.5.1 Fixed Time Length

As previously said, in the fixed time length category the whole duration of the signatures can be used. We create two protocols of different difficulty, as illustrated in Figure 3.13:

- **Intersession protocol**
  In this protocol, all the signatures contained in the *session 1* are included in the training set,

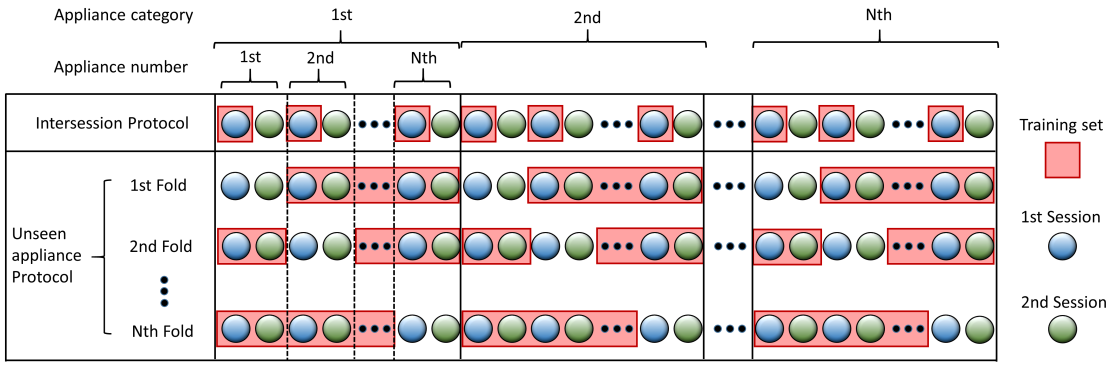


Figure 3.13: Difference between the *Intersession* and *Unseen Appliance* protocols. Signatures of *session 1* are shown in blue, signatures *session 2* are in green. The red squares underline the signatures included in the training set.

while the signatures of *session 2* compose the test set. The training and the test sets have the same size, i.e. 100 instances for ACS-F1 and 225 for ACS-F2. According to the protocol requirements, results have to be presented in the form of confusion matrices and total accuracy rates. The whole duration of the signatures (i.e. 1 hour) can be used. This protocol corresponds to a MS-ILM setup where the appliances are manually labeled "on site". All appliances of *session 2* are in the test set, then a total of 100 instances for ACS-F1 and 225 for ACS-F2 are tested.

- **Unseen Appliance protocol**
  In this protocol, all instances of both sessions are taken to perform a k-fold cross-validation. For the ACS-F1, k is set to 10 and for the ACS-F2, k is 15. The signatures are separated in order to have in every fold both instances of one appliance per category. Similarly to the previous protocol, results have to be presented in the form of confusion matrices and total accuracy rates averaged over the N folds. The whole duration of the signatures (i.e. 1 hour) can be used. All appliances in the database are tested one time, then a total of 200 instances for ACS-F1 and 450 for ACS-F2 are tested.

3.5.2 Variable Time Length

The *Intersession* and *Unseen Appliance* protocols presented above, allow to use the whole duration of the signals, namely 1 hour. For some application scenarios, such as building management systems, this duration is probably too large. In this case, shorter signature portions are more interesting and, for this reason, we propose a new protocol category, called *Variable Time Length*, which aims at evaluating the impact of the analysis window length. The goal is to measure the performance of the appliance identification task as a function of the window lengths. Such signals are obtained through a simple windowing operation moving a fixed length window on the signal and extracting the corresponding "sub-signatures" for further identification. The protocol proposes window lengths of 1 up to 60 minutes by 1 minute increments. The window length constraint is applied at testing time while, in the training phase, the length of the windows is free, up to the entire hour available in the training set.

We use the same structure and naming as for the *Fixed time length* protocols, but this time we add the term "dynamic":

- **Dynamic Intersession protocol**
  The train and test sets are split in the same way as for the *Intersession* protocol. The signatures in the *session 1* belong to the training set and those in the *session 2* to the test set. For the training phase, the whole duration of the signature up to 1 hour can be used. In the test phase the window length is increased from 1 minute up to 1 hour with a step of 1 minute. A total

of 60 accuracy rates are then computed, the first one for a duration of 1 minute, the second for a duration of 2 minutes, etc. The trend of the accuracy rate with the increasing window length has to be reported in a graph. The overlap between consecutive analysis windows is not specified in the current protocol definition.

- **Dynamic Unseen Appliance protocol**
  The train and test sets are split in the same way as the *Unseen Appliance* protocol: a k-fold cross validation is performed and every fold includes both instances of one appliance per category. As for the *Dynamic Intersession* protocol, the whole duration of the signature can be used at training time. In the test phase, the window length is increased from 1 minute up to 1 hour with a step of 1 minute. A total of 60 accuracy rates are then computed. As for the previous protocol, the trend of the accuracy rate with increasing window length has to be reported. The overlap between consecutive windows is not specified and can be chosen.

In Table 3.2 we report a summary of our protocols.

Type	**Fixed time length**		**Signal length impact**	
Protocol	Intersession	Unseen appliance	Dynamic Intersession	Dynamic Unseen appliance
Train set	Instances Session 1	k-folds on all session instances [a]	Instances Session 1	k-folds on all session instances [a]
Test set	Instances Session 2		Instances Session 2	
Train time	1 hour max. (whole instance)		1 hour max. (whole instance)	
Test time	1 hour max. (whole instance)		1 minute to 1 hour with an increment of 1 minute	
Result form	Confusion matrix, accuracy		Trend of accuracy rate when varying the window length	

[a] k=10 for the ACS-F1 and k=15 for the ACS-F2.

Table 3.2: Summary of the existing and proposed protocols for the ACS-F1 and ACS-F2 databases.

3.6 Classification

We apply three different machine learning techniques for the appliance recognition: K-NN, GMMs and HMMs. The two latter machine learning techniques are expected to be particularly interesting for this task given the state-based nature of the signals. More details on GMMs/HMMs library that we use for this and all our other works in this thesis are provided in the Appendix A.

3.6.1 K-NN algorithm

A K-NN classifier is based on the computation of distances between training and test data. A test data is classified by choosing the closest class among the k nearest neighbors. If $k = 1$, the test sample is simply assigned to the class of its nearest neighbor. If $k > 1$, then a majority voting algorithm is usually applied for determining the most represented class in the neighborhood. However, it can happen that multiple classes have the same number of nearest points among the k nearest neighbors (case of tie). In this case, the vote is usually attributed to the class having the lowest overall distortion. Another important aspect is the chosen distance metric, given that it is usually dependent on the nature of the feature space. In our case we test several distance metrics, as Euclidean, Manhattan, Chebychev, Minkowski, Mahalanobis, etc. to determine the one that best fits the identification problem. After some pre-tests, we decide to use the Euclidean distance.

For a feature vector sequence $X = \{x_1, \ldots, x_N\}$ we compute N independent decisions. In our implementation, we perform again a majority voting on the N decisions to determine the winning category. In other words, we compute a score S_i by accumulating the number of decisions in favor of category i for a given signature.

Baseline tests

	ACS-F1		ACS-F2	
	Intersession	Unseen Appliance	Intersession	Unseen Appliance
x_n	88.0 ($k = 2$)	44.0 ($k = 2$)	87.1 ($k = 1$)	49.8 ($k = 1$)
$x_n, \Delta x_n$	87.0 ($k = 1$)	46.0 ($k = 10$)	85.4 ($k = 1$)	51.8 ($k = 1$)
$x_n, \Delta x_n, \Delta\Delta x_n$	89.0 ($k = 1$)	51.0 ($k = 9$)	83.1 ($k = 1$)	53.6 ($k = 1$)

Table 3.3: Accuracy rates (%) for the baseline tests using K-NN.

We compute the accuracy rate of the identification task on both databases and we use the fixed time length protocols (*Intersession* and *Unseen appliance*). In order to evaluate the influence of the dynamic coefficients, we systematically compare three different cases: (i) using only the original coefficient (x_n), (ii) the original and velocity coefficients ($x_n, \Delta x_n$) and (iii) the original, velocity and acceleration coefficients ($x_n, \Delta x_n, \Delta\Delta x_n$). For every test we systematically increase the number of neighbors k and we observe the trend of the accuracy rate. We report in Table 3.3 the accuracy rate and between parentheses the number k of neighbors that gives the best result.

Depending on the protocol and the database under analysis, we obtain quite different results. However, in most of the cases the best results are obtained with $k = 1$. Higher values of k seem to introduce more instability of the scores which can intuitively be explained considering the high inter-class overlap leading to more frequent wrongly labeled nearest neighbors. Such results have also been observed in similar context where the 1-NN configuration apparently led to stable and consistent performances [81].

The inclusion of dynamic coefficients has different consequences depending on the protocol under analysis. The dynamic coefficients seem to deteriorate the accuracy rate when using the Intersession protocol. While for the ACS-F1 the results are ambiguous, for the ACS-F2 the accuracy rate drops by 4%. This effect can be explained by the fact that the information contained in the original coefficients is adequate if the appliance signatures have already been seen by classifiers. The inclusion of dynamic coefficients reveals more noisy than useful, yielding to a deterioration of the results. Therefore, we can state that for the Intersession protocol the information is mostly contained in the form of the signatures. The opposite case happens for the Unseen appliance protocol, where the inclusion of the dynamic coefficients yields an increment of approximately 4% for both the databases. The information contained in the dynamic coefficients is useful for the identification of appliances never seen before by classifiers. Intuitively, signatures belonging to the same category have different forms but similarities exist between their variations.

Weighted distance computation

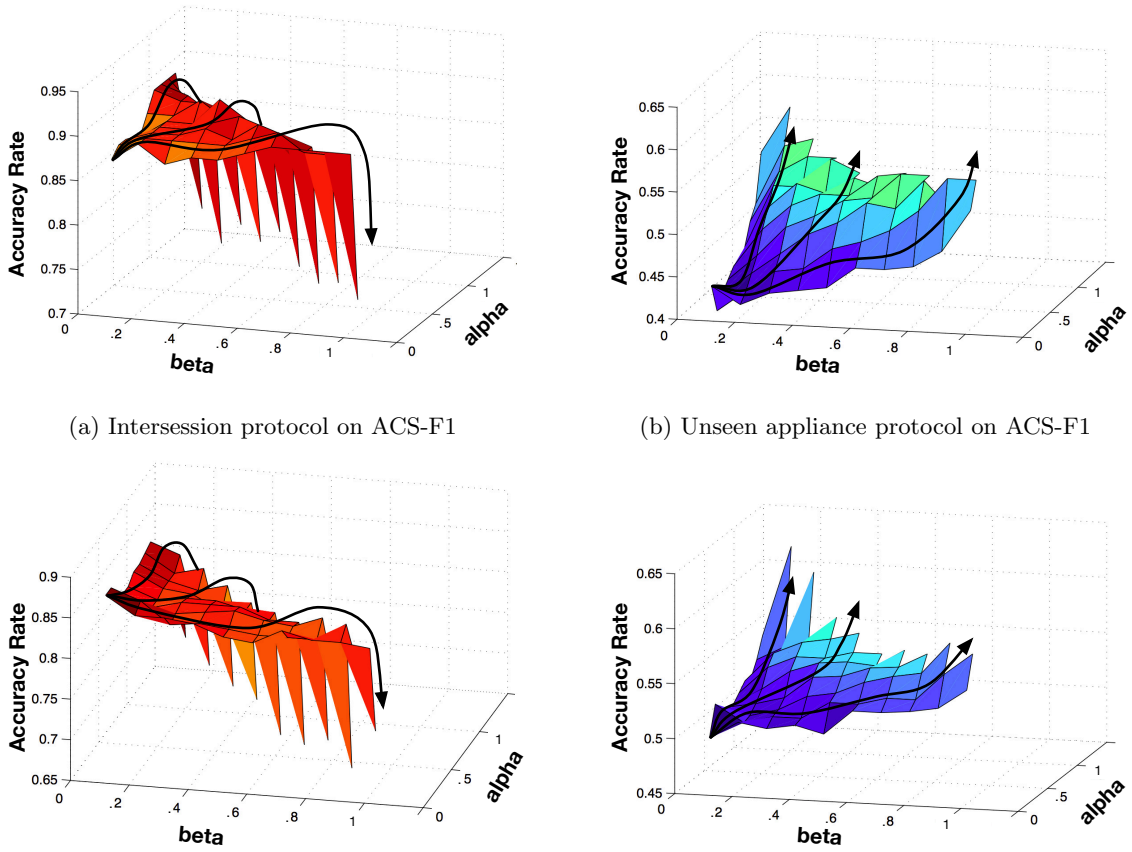
ACS-F1		ACS-F2	
Intersession	Unseen Appliance	Intersession	Unseen Appliance
92.0 [$\alpha = 0.4, \beta = 0.3$]	67.5 [$\alpha = 1, \beta = 0$]	87.1 [$\alpha = 0, \beta = 0$]	66.2 [$\alpha = 0, \beta = 1$]

Table 3.4: Best accuracy rates (%) using a weighted distance computation with a K-NN system.

In a previous work we have observed a benefit of using a weighted distance on the original and velocity coefficients on the ACS-F1 database [46]. We extend the same concept to the acceleration coefficients, by adding weighting parameters in the formula of the computation of the distance, as reported in the following equation:

$$dist(x_{ts}, x_{tr}) = (1 - \alpha - \beta)d(x_{ts}, x_{tr}) + \alpha d(\Delta x_{ts}, \Delta x_{tr}) + \beta d(\Delta\Delta x_{ts}, \Delta\Delta x_{tr}) \tag{3.7}$$

where d is the Euclidean distance, x_{ts} is a test feature vector and x_{tr} is a train feature vector. The parameters α and β change the contribution of the original, velocity and acceleration coefficients. We vary α and β between 0 and 1 with a step of 0.1 and we constrain their sum to be less than or equal to 1.



(a) Intersession protocol on ACS-F1

(b) Unseen appliance protocol on ACS-F1

(c) Intersession protocol on ACS-F2

(d) Unseen appliance protocol on ACS-F2

Figure 3.14: Profiles of the accuracy rates using a K-NN and when varying the α and β parameters respectively weighting the dynamic coefficients.

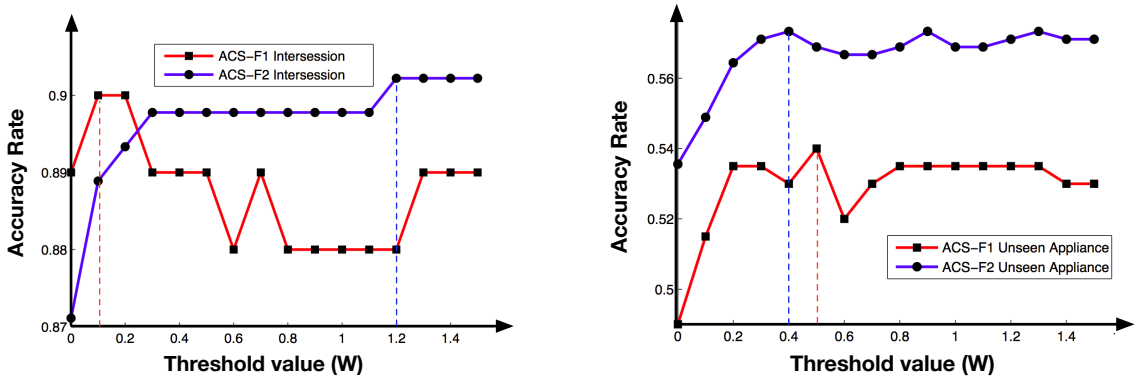
In Figure 3.14 we report the profiles of the accuracy rate when varying the α and β parameters and in Table 3.4 the best accuracy rates. By observing the profiles, we notice two different behaviors depending on the type of protocol: for the Intersession the results seem to deteriorate when increasing the α and β parameters, while the opposite occurs for the Unseen appliance protocol. The trend of the profiles confirms what we have observed during the baseline tests: the Intersession protocol finds useful information in the form of the signatures while the Unseen appliance protocol yields better results when using the dynamic coefficients capturing temporal variations of the signatures.

Thresholding

ACS-F1		ACS-F2	
Intersession	Unseen Appliance	Intersession	Unseen Appliance
90.0 [$T_P = 0.1$]	54.0 [$T_P = 0.5$]	90.2 [$T_P = 1.2$]	57.3 [$T_P = 0.4$]

Table 3.5: Best accuracy rates (%) using a thresholding procedure with a K-NN system.

In this test we evaluate the effect of applying power thresholding, namely eliminating from the signatures the observations where the value of the active power is below a given threshold T_P. Intuitively, this is related to the fact that appliances are difficult to discriminate when they are consuming a small quantity of energy, e.g. when they are off or in stand-by. Several signatures have many observations near 0, independently of the type of appliances and this fact can potentially introduce noisy results. During this experiment we include the dynamic coefficients. As illustrated in Figure 3.15, we tune



(a) Intersession on ACS-F1 and ACS-F2

(b) Unseen Appliance on ACS-F1 and ACS-F2

Figure 3.15: Trend of the accuracy rate when varying the threshold T_P.

	Intersession protocol										**Unseen appliance protocol**									
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer
Hifi	.9	0	.1	0	0	0	0	0	0	0	.85	0	.1	0	0	0	0	0	0	.05
Television	0	.9	0	0	0	.1	0	0	0	0	0	.65	0	.05	0	0	0	.05	0	.25
Mobile P.	0	0	.8	0	0	.2	0	0	0	0	0	0	.95	0	0	.05	0	0	0	0
Coffee M.	0	0	0	1	0	0	0	0	0	0	.1	0	.1	.75	0	.05	0	0	0	0
Computer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	.65	0	0	.35	0	0
Fridge	0	0	0	0	0	1	0	0	0	0	0	0	.2	0	0	.8	0	0	0	0
Lamp	0	0	.1	0	0	0	.9	0	0	0	.15	0	.3	0	0	.05	.4	0	.05	.05
Laptop	0	0	0	0	.1	0	0	.7	0	.2	.1	0	0	0	.1	0	0	.75	0	.05
Oven	0	0	0	0	0	0	0	0	1	0	.05	0	.15	.15	.15	.1	0	0	.4	0
Printer	0	0	0	0	0	0	0	0	0	1	0	.1	.25	.05	0	0	0	.05	0	.55

Table 3.6: In the left side, the confusion matrix of the Intersession protocol on the ACS-F1 database for a K-NN with thresholding. In the right side, the confusion matrix of the Unseen appliance protocol on the ACS-F1 database for a K-NN with weighted distances computation.

the threshold T_P making it range from 0 to 1.5 W, with a step of 0.1 W. The best results in terms of accuracy rate have been reported in Table 3.5. In all the cases we have better performances compared to the baseline results, however, the most important improvement is achieved by the Intersession protocol on the ACS-F2.

Comments on the results

For the *Intersession* protocol on the ACS-F1 database the baseline accuracy rate is 89% using all the dynamic coefficients when $k = 1$. However, the influence of the dynamic coefficients is not particularly important, given the small improvement compared to the original coefficients (88% using $k = 2$). The weighted distance computation improves the accuracy rate of 3%, however, observing the profile of the weighted distances in Figure 3.14a, we notice that the profile of the accuracy rate is noisy, maybe due to the low quantity of data. The thresholding yields an improvement of 1% when T_P is equal to 0.1 W. The confusion matrix of the best K-NN system is reported in Table 3.6 (left). With the exception of Laptop and Mobile phone, all categories attain the 90% of accuracy rate. Several categories obtain an accuracy rate of 100%.

For the *Intersession* protocol on the ACS-F2 database the baseline accuracy rate is 87.1%. The inclusion of dynamic coefficients deteriorates the results, in particular the velocity coefficients make drop the accuracy rate to 85.4% and the acceleration coefficients to 83.1%. The profile of the weighted distances in Figure 3.14c is similar to the one of the ACS-F1 database in Figure 3.14a, however it appears smoother. When computing the weighted distance, the best case is obtained with $\alpha = 0$ and

	Intersession protocol															Unseen appliance protocol													
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver
Hifi	.93	0	.07	0	0	0	0	0	0	0	0	0	0	0	0	.77	0	.03	0	0	.03	0	0	0	0	.07	0	0	.1	0
Television	0	.93	0	0	0	0	0	.07	0	0	0	0	0	0	0	.03	.53	0	.07	0	0	0	.13	0	0	0	0	0	.23	0
Mobile P.	0	0	.8	0	0	0	0	0	0	0	0	0	0	0	.2	.13	0	.5	0	0	.03	.03	0	0	0	.03	.07	.07	0	.13
Coffee M.	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.07	0	.03	.87	0	0	0	0	0	0	0	.03	0	0	0
Computer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.07	0	0	.8	0	0	.13	0	0	0	0	0	0	0
Fridge	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	.03	0	.03	0	0	.67	0	0	.07	0	.1	.1	0	0	0
Lamp Inc.	0	0	0	0	0	0	.93	0	0	0	.07	0	0	0	0	.2	0	0	0	0	0	.63	0	.03	0	0	.1	.03	0	0
Laptop	0	.07	0	0	0	0	0	.73	0	.07	.13	0	0	0	0	.03	.03	0	0	.23	0	0	.5	0	0	0	0	0	.2	0
Oven	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	.1	0	0	0	.8	.07	0	.03	0	0	0
Printer	0	0	.07	0	.07	0	0	.07	0	.8	0	0	0	0	0	.07	.07	.1	.1	0	.03	.03	.03	.03	.23	0	.13	0	.13	.03
Fan	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	.07	0	0	0	0	.03	.1	.03	.03	0	.6	.07	.07	0	0
Kettle	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Lamp CFL	0	.07	0	0	0	0	0	.07	0	.13	0	0	.73	0	0	.07	0	.03	0	0	0	0	0	0	0	0	.07	.8	0	.03
Monitor	0	0	0	0	0	0	0	.07	0	0	0	0	0	.93	0	.07	0	.03	.03	.03	0	0	.03	0	.03	0	0	0	.77	0
Shaver	0	0	.27	0	0	0	0	0	0	0	0	0	0	0	.73	.17	0	.03	0	0	.03	0	0	0	0	.03	.2	.07	0	.47

Table 3.7: In the left side, the confusion matrix for the Intersession protocol on the ACS-F2 database using K-NN with thresholding. In the right side, the confusion matrix for the Unseen appliance protocol on the ACS-F2 database using K-NN with weighted distances computation.

$\beta = 0$, namely when the dynamic coefficients are not included at all. Also in this case the thresholding yields an improvement of the accuracy rate, in particular it attains 90.2% when T_P is set to 1.2 W. The confusion matrix of the best case is reported in Table 3.7 (left). All categories attain at least 73% in terms of accuracy rate. The greatest misclassification errors are between Shaver - Mobile phone, Laptop - Fan and Lamp CFL - Printer. Several categories obtain an accuracy rate of 100%.

For the *Unseen appliance* protocol on the ACS-F1 database the best baseline accuracy rate is 51%. The inclusion of the dynamic coefficients is beneficial in this case, with the accuracy rate going from 44% to 51%. The profile of the weighted distance computation, depicted in Figure 3.14b, shows the importance of dynamic coefficients: the less the weight of the original coefficients, the better is the accuracy rate up to 67.5%. The thresholding seems to slightly improve the accuracy rate when $T_P = 0.5W$. The confusion matrix of the best case is reported in Table 3.6 (right). Some categories, as Lamp and Oven, show poor performances (40%). The greatest misclassification errors are between Computer - Laptop, Lamp - Mobile phone, Television - Printer and Printer - Mobile phone.

For the *Unseen appliance* protocol on the ACS-F2 database the best overall accuracy rate is 53.6% with $k = 1$. As in the previous case, the inclusion of the dynamic coefficients is particularly beneficial, improving the scores of almost 4%. The profile of the weighted distances in Figure 3.14d is similar to the one in Figure 3.14b. In fact, as in the previous case, the less the weight of the original coefficients, the better is the accuracy rate. When using the weighted distance computation the accuracy rate is 66.2%. Compared to the weighted distance, the thresholding operation provides a smaller improvement on the accuracy rate, i.e. 57.3% when $T_P = 0.4W$. The confusion matrix of the best case is reported in Table 3.7 (right). Kettle is the only category attaining the 100% of accuracy rate. Printer provides the worst performances (23%), followed by Shaver, Laptop, Mobile phone and Television. The greatest misclassification errors are between Television - Monitor, Laptop - Monitor and Laptop - Computer.

Discussion

Generally speaking, the results provided by the K-NN algorithm are consistent between databases and protocols.

- When using the Intersession protocol, the inclusion of the dynamic coefficients provides slight improvements on the ACS-F1 database and sensibly deteriorates the accuracy rate on the ACS-F2. By observing the profiles of the weighted distance computations, the accuracy rates decrease when increasing the contribution of the dynamic coefficients. For the ACS-F1 the weighted distance profile is not as smooth as the one of the ACS-F2, probably due to the poor quantity of data. Generally speaking, adding dynamic coefficients seems to introduce more variability than

useful information for the Intersession protocol and therefore it deteriorates the results. For the Intersession protocol the useful information is mostly related to the values of the original coefficients. This can be explained by the fact that the classifiers have already seen the appliances during the training and the original coefficients better discriminate the appliance categories than the dynamic ones.

- When using the Unseen appliance protocol, the inclusion of the dynamic coefficients provides significant improvements on both databases. By observing the profiles of the weighted distance computations, the less the weight of the original coefficients, the better is the accuracy rate. The information contained in the dynamic coefficients is useful for identifying appliances never seen before by classifiers. Intuitively the signatures belonging to the same category but different appliances can be very different, however similarities between variations of these signatures exist.
- The thresholding improves the accuracy rate independently of the database and the protocol. Moreover we notice a significant improvement for the Intersession protocol on the ACS-F2 database.

In conclusion, when using the K-NN algorithm, the dynamic coefficients seem to be particularly important for the Unseen appliance protocol, while the opposite happens for the Intersession protocol. The thresholding operation yields to better results in all the cases, however the importance of the improvement depends on the type of database and protocol.

3.6.2 Gaussian Mixture models

The instances in the test set are checked against the models and a *likelihood* score is computed for every model. Using the *likelihood* score and the *a priori* probability, we compute the *a posteriori* probability, according to the Bayes' law. In our databases the appliances are uniformly distributed, therefore we have equal priors.

Baseline tests

	ACS-F1		ACS-F2	
	Intersession	Unseen Appliance	Intersession	Unseen Appliance
x_n	91.0 ($K = 16$)	60.0 ($K = 4$)	92.8 ($K = 16$)	62.9 ($K = 8$)
$x_n, \Delta x_n$	95.2 ($K = 28$)	70.5 ($K = 4$)	94.8 ($K = 40$)	71.1 ($K = 8$)
$x_n, \Delta x_n, \Delta\Delta x_n$	94.5 ($K = 40$)	73.5 ($K = 4$)	94.0 ($K = 32$)	72.2 ($K = 8$)

Table 3.8: Accuracy rate (%) of the baseline tests using GMMs.

As for the K-NN, our baseline tests analyze the influence of the dynamic coefficients. In our settings, we use a K-Means for initializing the Gaussian mixtures and the EM procedure. The random initial attribution of the data points of the K-Means can make the EM converge to different final parameters of the Gaussian. For this reason, we perform the computation of the models 10 times, randomly changing the initial conditions, and averaging the performances. As for the K-NN, we compute the accuracy rates on both databases and protocols. We systematically compare (i) the original coefficient (x_n), (ii) the original and velocity coefficients ($x_n, \Delta x_n$) and (iii) the original, velocity and acceleration coefficients ($x_n, \Delta x_n, \Delta\Delta x_n$). For every test we vary the number of Gaussians K from 1 to 40 with steps of 4 Gaussians. In Table 3.8 we report the best accuracy rate and the number of Gaussians.

In Figure 3.16 we show the relation between the models and the log-likelihood scores when classifying a signature. The yellow line is the log-likelihood obtained when testing the Fridge model against a Fridge signature. All the other lines are the competing models tested against the same signature. We separate the figure in three parts: (i) the original coefficients, (ii) the velocity coefficients and (iii) the acceleration coefficients. When using the original coefficients, the log-likelihood slowly decreases for the Fridge model while it strongly decreases for all the other models. We align the log-likelihoods with the active power and we notice that the un-matching models constantly decrease their log-likelihoods. The slope of the log-likelihood decrements depending on the difference between the Fridge mode, i.e.

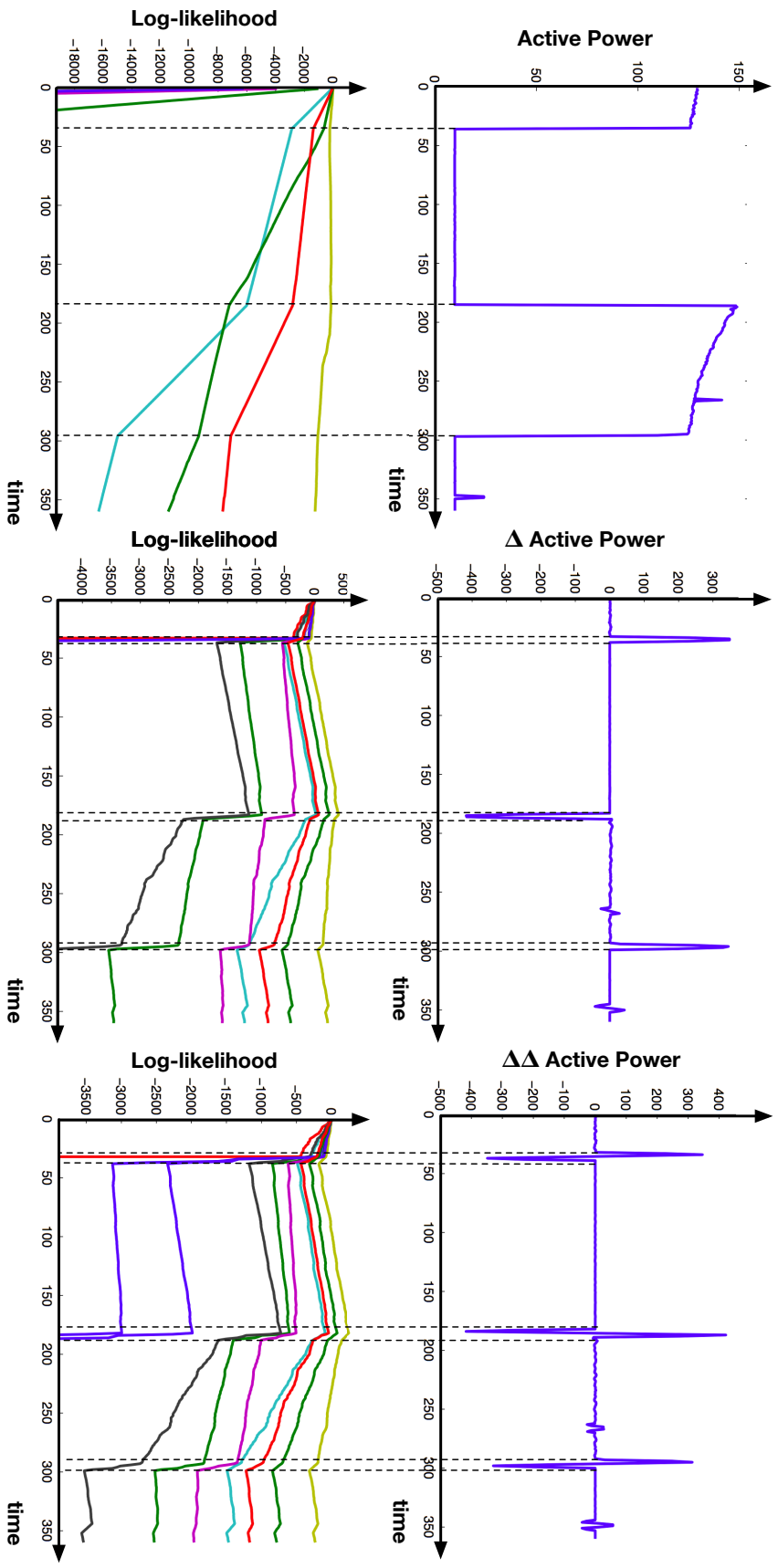
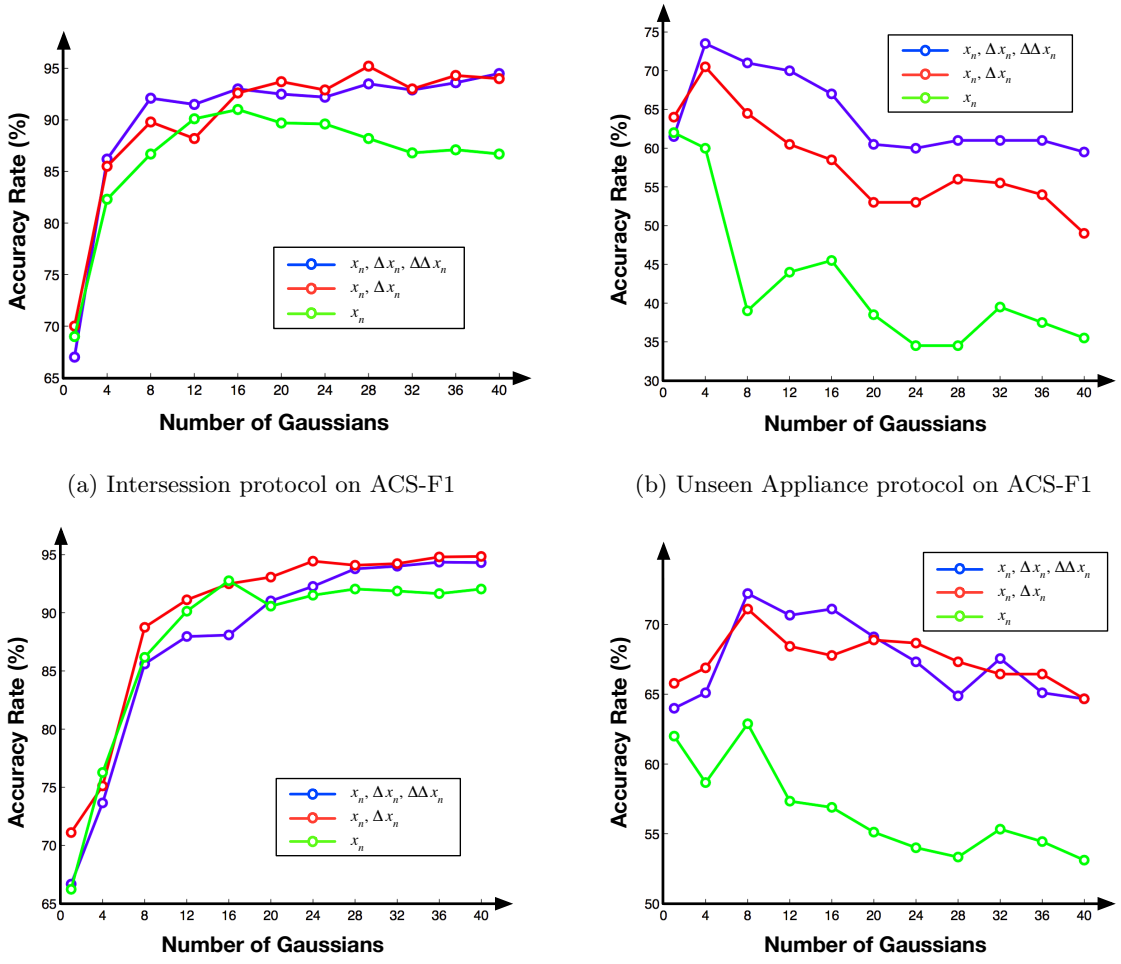


Figure 3.16: Relation between the models and the log-likelihood scores obtained with GMMs on a Fridge signature. On the upper part, the active power, the velocity and the acceleration coefficients are plot as a function of time. On the bottom part, the cumulated log-likelihood is plot as a function of time. The yellow line corresponds to the matching fridge model, i.e. the model that is supposed to win. The other lines correspond to the competing un-matching models.



(a) Intersession protocol on ACS-F1

(b) Unseen Appliance protocol on ACS-F1

(c) Intersession protocol on ACS-F2

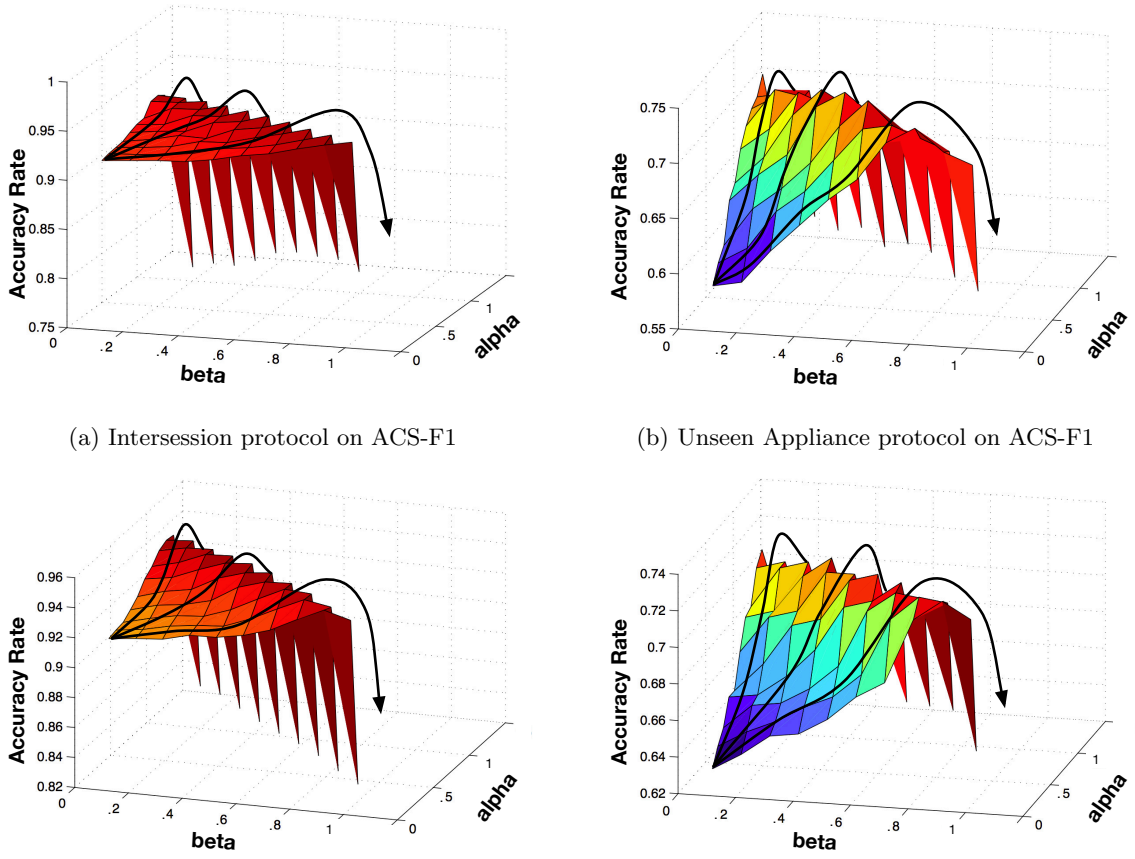
(d) Unseen Appliance protocol on ACS-F2

Figure 3.17: Evolution of the accuracy rate as a function of the number of Gaussians K for GMMs.

compression - non compression phase and the modes of the un-matching models. For the velocity and acceleration coefficients the situation is very different. All the models seem to provide the same log-likelihood when there are no variations in the signatures. When some variations are verified, the log-likelihood of the Fridge model slowly decreases, while it dramatically drops for almost all the other models.

The evolution of the accuracy of the GMMs as a function of the number of mixtures is depicted in Figure 3.17. For the Intersession protocol, the accuracy rate is improving when increasing the number of Gaussians, showing the ability of higher order GMMs to better capture the details of the probability density functions. The performances increase significantly until 8-12 mixtures where the improvements start saturating. For the Unseen appliance protocol, the accuracy rate improves up to a particular number of Gaussians and after the results seems to decrease. This observation is probably related to the overfitting, because the model is excessively complex and it loses the capacity to generalize to test data.

The inclusion of dynamic coefficients is beneficial independently of the protocol used. However, for the Intersession protocol, the acceleration coefficients do not seem to improve results. In most situations, the accuracy is higher when the dynamic coefficients are included, showing the benefit of using such coefficients.



(a) Intersession protocol on ACS-F1

(b) Unseen Appliance protocol on ACS-F1

(c) Intersession protocol on ACS-F2

(d) Unseen Appliance protocol on ACS-F2

Figure 3.18: Profiles of the accuracy rates when varying the α and β parameters using GMMs.

ACS-F1		ACS-F2	
Intersession	Unseen Appliance	Intersession	Unseen Appliance
95.4 $[\alpha = 0.1, \beta = 0.8]$	75.0 $[\alpha = 0.3, \beta = 0.4]$	95.2 $[\alpha = 0.5, \beta = 0.4]$	73.1 $[\alpha = 0, \beta = 0.8]$

Table 3.9: Results of the weighted probability computation using GMMs.

Weighted probability computation

As for K-NN, we weight the contribution of the original, velocity and acceleration coefficients through the α and β parameters. We compute a weighted probability (p_{tot}):

$$p_{tot} = (1 - \alpha - \beta)p(x_n|M_j) + \alpha p(\Delta x_n|M_j) + \beta p(\Delta\Delta x_n|M_j) \tag{3.8}$$

where $p(x_n|M_j)$, $p(\Delta x_n|M_j)$ and $p(\Delta\Delta x_n|M_j)$ are respectively the likelihoods of the original feature vector x_n, the velocity coefficients Δx_n and the acceleration coefficients $\Delta\Delta x_n$ given a category M_j. We vary α and β between 0 and 1 with a step of 0.1 and we constrain their sum to be less than or equal to 1. With the weighted probability computation, we observe a systematic improvement for all protocols and databases in comparison to the baseline results of Table 3.8.

In Figure 3.18 we show the profile of the weighted probability computation using different databases and protocols, while in Table 3.9 we report the best cases. We notice that the computation of the weighted probability is beneficial for all the databases and protocols, however the improvements are slight, i.e. approximately 1-1.5%.

	Intersession protocol										Unseen appliance protocol									
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer
Hifi	.97	.02	0	0	0	0	0	0	0	.01	.65	0	0	0	0	.15	.1	.05	0	.05
Television	0	1	0	0	0	0	0	0	0	0	0	.7	0	.1	0	0	.05	.1	0	.05
Mobile P.	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Coffee M.	0	0	0	1	0	0	0	0	0	0	0	0	0	.75	0	0	0	0	.2	.05
Computer	0	0	0	0	1	0	0	0	0	0	0	.15	0	0	.7	0	0	.15	0	0
Fridge	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	.9	0	0	.1	0
Lamp	0	0	0	0	0	.03	.88	0	.09	0	0	.1	.1	0	.1	.1	.35	0	.05	.2
Laptop	0	.09	0	0	.1	.03	0	.7	0	.08	0	0	0	0	0	0	.1	.9	0	0
Oven	0	0	0	0	0	0	0	0	1	0	0	0	0	.1	0	0	0	0	.75	.15
Printer	0	0	0	.01	0	0	0	0	0	.99	.05	0	0	.05	0	.05	.05	0	0	.8

Table 3.10: In the left side, the confusion matrix of the Intersession protocol for GMMs with weighted probability computation. In the right side, the confusion matrix of the Unseen appliance protocol for GMMs with weighted probability computation.

	Intersession protocol														Unseen appliance protocol													
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver
Hifi	.89	.06	0	.03	0	0	.01	0	0	.01	0	0	0	0	0	.63	0	.03	0	0	.17	0	.07	.03	.03	0	0	.03	0	0
Television	0	.99	0	0	0	.01	0	0	.01	0	0	0	0	0	0	0	.7	0	.07	0	.13	0	.03	0	.07	0	0	0	0	0
Mobile P.	0	0	.78	0	0	0	0	0	0	.05	0	0	0	0	.17	0	0	.87	0	0	0	0	0	0	0	0	0	0	0	.13
Coffee M.	0	0	0	.93	0	0	0	0	.07	0	0	0	0	0	0	0	0	0	.7	0	0	0	0	.17	.13	0	0	0	0	0
Computer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.07	0	0	.7	.07	0	.1	.03	.03	0	0	0	0	0
Fridge	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	.07	0	.87	0	0	0	0	.07	0	0	0	0
Lamp Inc.	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.03	0	.1	.53	0	0	0	.33	0	0	0	0
Laptop	0	.11	0	0	.01	0	0	.85	0	.02	0	0	0	0	0	0	.07	0	0	.2	0	0	.67	0	0	0	0	.03	.03	0
Oven	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	.13	0	0	.8	.07	0	0	0	0	0
Printer	0	0	0	.06	0	0	0	0	.01	.93	0	0	0	0	.01	.07	0	.07	.1	0	.07	0	0	.07	.5	0	0	0	0	.13
Fan	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	.07	0	0	0	.07	0	0	0	0	.87	0	0	0	0
Kettle	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Lamp CFL	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	.03	0	.03	0	0	.03	0	.1	0	.77	0	.03
Monitor	0	.02	0	0	0	0	0	.05	0	0	0	0	0	.93	0	0	0	0	0	0	0	0	.17	0	0	0	0	0	.83	0
Shaver	.01	0	.01	0	0	0	0	0	0	0	0	0	0	0	.99	.13	0	.23	0	0	.03	0	0	0	0	0	0	.07	0	.53

Table 3.11: In the left side, the confusion matrix for Intersession protocol on the ACS-F2 using GMMs with weighted probability computation. In the right side, the confusion matrix for Unseen appliance protocol on the ACS-F2 using GMMs with weighted probability computation.

Comments on the results

For the *Intersession* protocol on the ACS-F1 database the baseline accuracy rate is 94.5%. The results is obtained using all the dynamic coefficients when $K = 40$. The influence of the velocity coefficients yields to an increment of more than 4%. The inclusion of the acceleration coefficients slightly deteriorates the results. The weighted probability computation slightly improves the accuracy rate, as illustrated by the profile in Figure 3.18a. The confusion matrix of the best case is reported in Table 3.10 (left). As for the K-NN, the Laptop is the most difficult category to recognize. All the other categories attain at least 88% of accuracy rate and several others obtain an accuracy rate of 100%.

For the *Intersession* protocol on the ACS-F2 database the baseline accuracy rate is 94%. As for the ACS-F1 database, the influence of the velocity coefficients is very beneficial while the acceleration coefficients do not further improve the results. As illustrated in Figure 3.18c, also for the ACS-F2 database the weighted probability computation slightly improves the accuracy rate (95.2%). The confusion matrix is reported in Table 3.11 (left). With the exception of Mobile phone, all the categories attain at least 85% in terms of accuracy rate. The misclassification between Mobile phone - Shaver is also present for GMMs, and others, as Laptop - Television, have an important impact. Several categories attain an accuracy rate of 100%.

For the *Unseen appliance* protocol on the ACS-F1 database the best baseline accuracy rate is 73.5%. As for the K-NN, the inclusion of the dynamic coefficients is particularly beneficial, increasing the accuracy rate of 13.5%. In Figure 3.18b the profile of the weighted probability computation shows the importance of the dynamic coefficients. With the tuning of the α and β parameters we improve the results up to 75%. The confusion matrix is reported in Table 3.10 (right). As for the K-NN, in this case the Lamp category provides poor performances (35%). This is due to the great variations between appliances belonging to the Lamp category, mostly due to the light bulb power differences, and making difficult the realization of a generic model. The greatest misclassification errors are between Lamp - Printer and Coffee Machine - Oven.

For the *Unseen appliance* protocol on the ACS-F2 database, the best baseline accuracy rate is 72.2%. As for the ACS-F2, the inclusion of the dynamic coefficients is particularly beneficial, with an improvement of approximately 10%. In Figure 3.18d the weighted probability computation is depicted. Again, the tuning of the α and β parameters improves the results, arriving in this case up to 73.1%. The confusion matrix is reported in Table 3.11 (right). As for the K-NN, Kettle is the only category performing well (100%) and Printer provides the worst performances (50%). As the Lamp in the previous case, Printer is another category that can be difficultly generalized, because of the great differences between appliances belonging to this category. Shaver and Lamp Inc. are also difficult to be correctly classified. The greatest misclassification errors are between Lamp Inc. - Fan, Shaver - Mobile phone and Laptop - Computer. As show in Figure 3.11, the misclassified categories present close clusters in the P-Q plane.

Discussion

Generally speaking, the results provided by GMMs are coherent with those obtained with the K-NN. However, GMMs perform better than K-NN, yielding an improvement of about 3-4% for the *Intersession* protocol and about 7-8% for the *Unseen appliance* protocol. A more exhaustive comparison will be later presented.

Focusing on GMMs, we report the following observations:

- when using the Intersession protocol, the inclusion of the velocity coefficients provides significant improvements, while the acceleration coefficients do not seem to be particularly effective. In particular, when using only the original coefficients, a fewer number of Gaussians tends to saturate the results (about 16 for both the databases). When adding the dynamic coefficients, a higher number of Gaussians is required (around 30-40).

- when using the Unseen appliance protocol, the inclusion of the dynamic coefficients provides significant improvements on both databases. Moreover, when using this protocol, a fewer number of Gaussians is needed before overtraining the classifiers.

- the weighted probability computation is beneficial for all the databases and protocols, showing the importance of the dynamic coefficients when using GMMs. All the profiles have a similar trend: when increasing the contribution of the dynamic coefficients, the accuracy rate increases up to a peak. However, if the original coefficients have a too small weight, then the accuracy rate starts to decrease. The difference between the starting value (i.e. using only the original coefficients) and the best accuracy rate is smaller for the Intersession protocol than for the Unseen appliance protocol.

3.6.3 Hidden Markov models

HMMs are supposed to work efficiently with electrical appliances, given the state-base nature of the electric signatures. In our implementation each state of models corresponds to a real state of the appliance. For instance, in Figure 3.19 we plot all the samples contained in the Fridge category of the ACS-F2 database on the P-Q plane. Different colors have been used for representing different states: the non-compression phase is in green, the opening of the door in blue and the compression phase in red. By using the Viterbi algorithm we recover the path of states, i.e. the *alignment*, and its probability, i.e. the *likelihood*. As for the GMMs, the *likelihood* score and the *a priori* probability are used for computing the *a posteriori* probability, according to the Bayes' law.

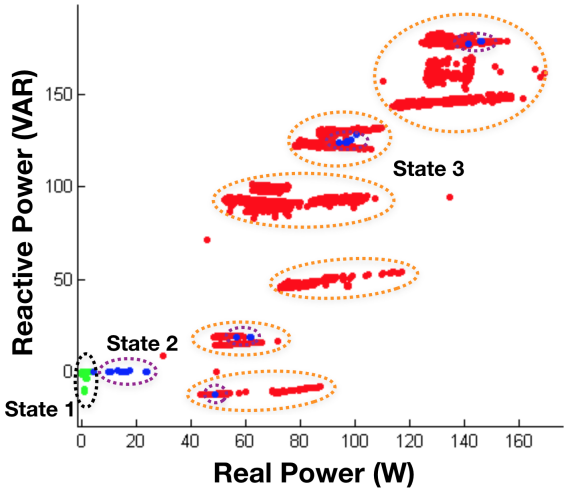


Figure 3.19: All samples contained in the Fridge category of the ACS-F2 database are plot on the P-Q plane. The non-compression phase is in green, the opening of the door in blue and the compression phase in red.

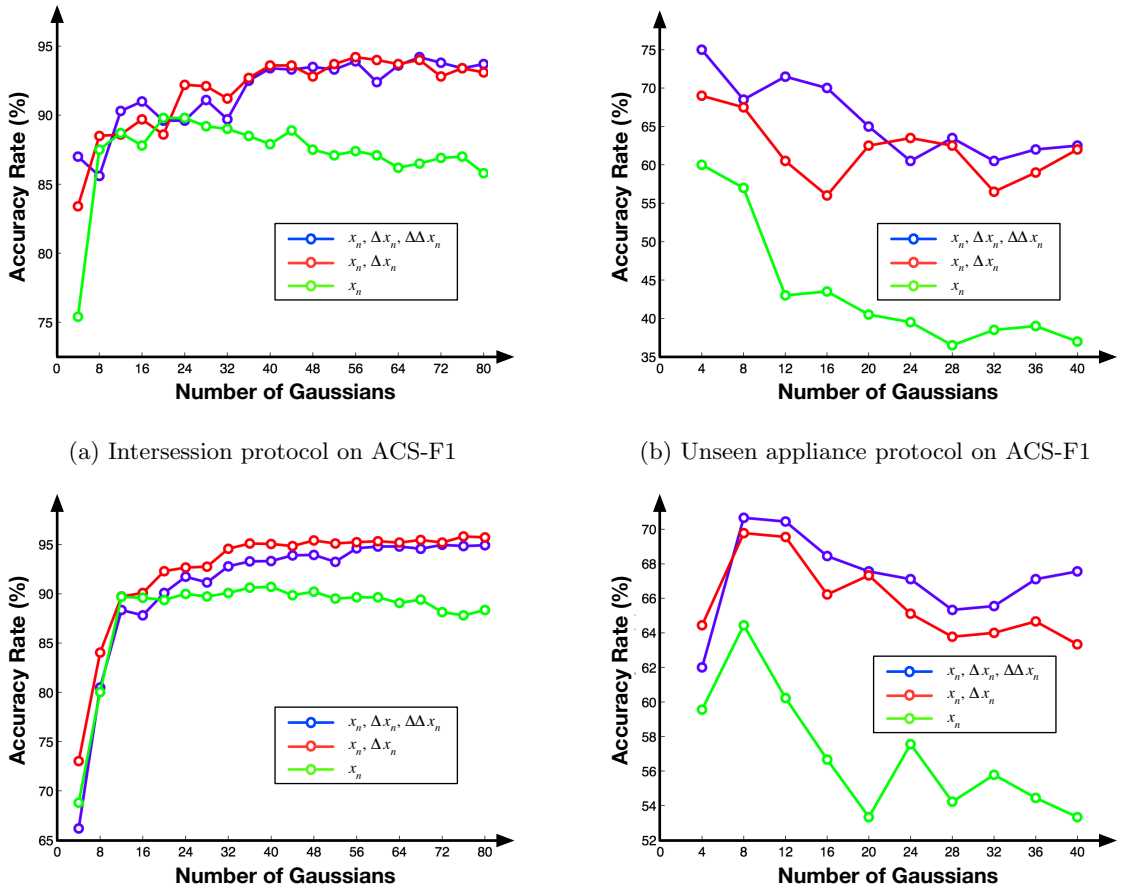
The HMMs structure is usually created by using information on the nature of the model or can be computed with complex strategies, mainly in the information theory field [82]. We recover the ground truth by manually finding the states in each signature. This task is simple for some categories having a clear distinction between the states, e.g. Kettle and Lamp, while is difficult for others having complex electrical consumption trends in time, e.g. Computer Station and Laptop. This annotation procedure is clearly prone to errors, potentially impacting the recognition rate. For this reason, we opt for simple models with two states for almost all the categories, representing respectively the on/off states. For the Fridge category we use three states to represent the compression/non-compression phase/door opening. All the models are ergodic. The ground truth of states has been used as starting point for the computation of the HMMs parameters. During the training phase, the HMMs iteratively find the most probable sequence of states that can converge to a different solution than the starting point.

Baseline tests

Similarly to the GMMs, we use a K-Means for initializing the Gaussian mixtures and the EM procedure. Given the random initial attribution of the data points of the K-Means, we perform the computation of the models 10 times, randomly changing the initial conditions, and averaging the performances. As for the K-NN and GMMs, we compute the accuracy rate for both databases and protocols, systematically comparing (i) the original coefficient (x_n), (ii) the original and velocity coefficients ($x_n, \Delta x_n$) and (iii) the original, velocity and acceleration coefficients ($x_n, \Delta x_n, \Delta\Delta x_n$). For every test we vary the total number of Gaussians K from 4 to 80 with steps of 4. The Gaussians are equally distributed among the states.

The evolution of the accuracy rate of HMMs as a function of the number of mixtures is depicted in Figure 3.20. The trend of the accuracy rate is very similar to that of GMMs for both the Intersession and Unseen appliance protocols. In the first case the accuracy rate is improving when increasing the number of Gaussians. The performances are increasing significantly until 50-60 mixtures where the improvements start saturating. In the second case the accuracy rate is improving up to a small number of Gaussians and after that the results decrease probably due to overfitting, as previously described.

In Table 3.12 we report the best accuracy rate and the number of Gaussians. The inclusion of dynamic coefficients is beneficial independently of the protocol used. However, for the Intersession protocol, the acceleration coefficients do not seem to improve results. In most situations, the accuracy is higher when the dynamic coefficients are included, showing the benefit of using such coefficients.



(a) Intersession protocol on ACS-F1

(b) Unseen appliance protocol on ACS-F1

(c) Intersession protocol on ACS-F2

(d) Unseen appliance protocol on ACS-F2

Figure 3.20: Evolution of the accuracy rate as a function of the number of Gaussians K for HMMs.

	ACS-F1		ACS-F2	
	Intersession	Unseen appliance	Intersession	Unseen appliance
x_n	89.8 ($K = 20$)	60 ($K = 4$)	90.7 ($K = 40$)	64.4 ($K = 8$)
$x_n, \Delta x_n$	94.2 ($K = 56$)	69 ($K = 4$)	95.8 ($K = 76$)	69.8 ($K = 8$)
$x_n, \Delta x_n, \Delta\Delta x_n$	94.2 ($K = 68$)	74.5 ($K = 4$)	95.0 ($K = 72$)	72.4 ($K = 8$)

Table 3.12: Accuracy rates (%) of the baseline tests using HMMs.

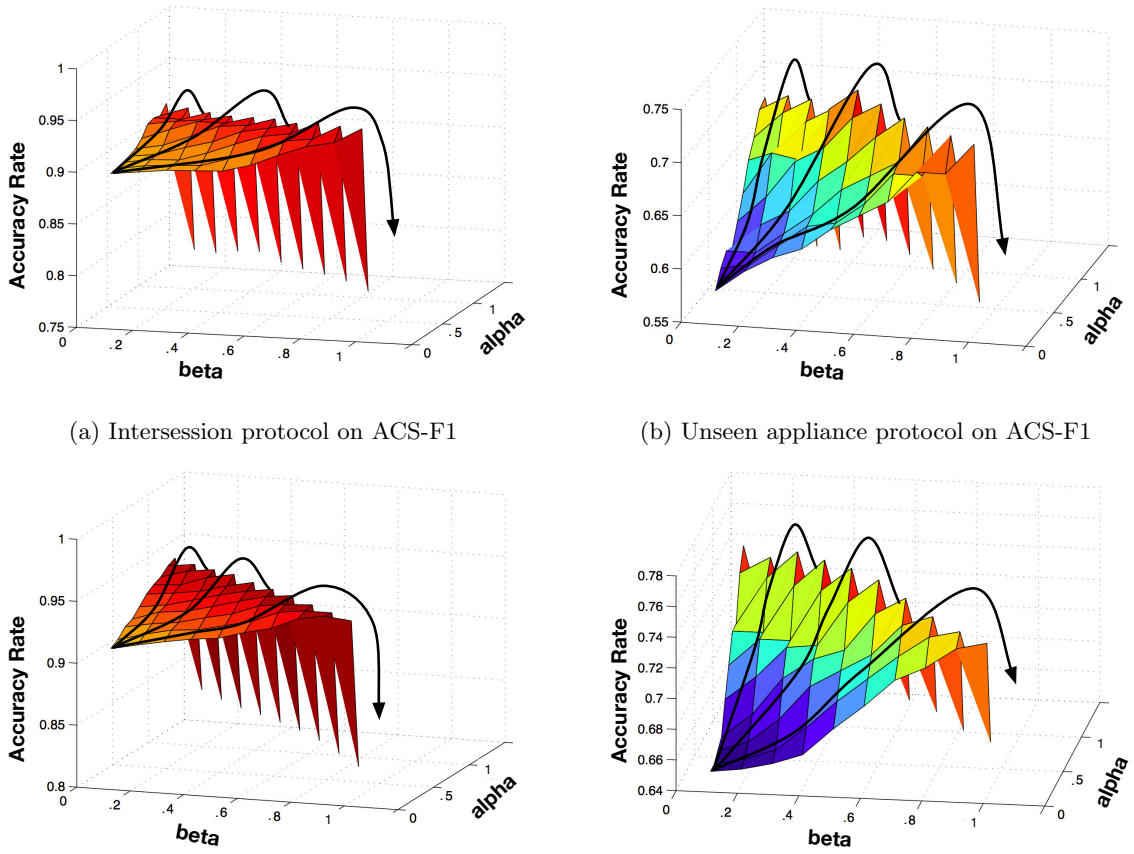
Weighted probability computation

We compute a weighted probability p_{tot} using the Formula 3.8. In Table 3.13 we report the best values when tuning the α and β parameters.

Generally speaking, we notice a slight increment of the performances compared to the baseline tests using all the coefficients. However, for the Unseen appliance protocol on the ACS-F2 we observe an interesting improvement of almost 4%.

ACS-F1		ACS-F2	
Intersession	Unseen appliance	Intersession	Unseen appliance
95.2 ($\alpha = 0, \beta = 0.9$)	74.5 ($\alpha = 0.5, \beta = 0.4$)	95.2 ($\alpha = 0.6, \beta = 0.3$)	76.2 ($\alpha = 0.6, \beta = 0.3$)

Table 3.13: Best accuracy rates (%) for the weighted probability computation using HMMs.



(a) Intersession protocol on ACS-F1

(b) Unseen appliance protocol on ACS-F1

(c) Intersession protocol on ACS-F2

(d) Unseen appliance protocol on ACS-F2

Figure 3.21: Profiles of the accuracy rates when varying the α and β parameters using HMMs.

State Recognition

The *alignment* can be compared with the original labels for obtaining an accuracy rate for state recognition. The *alignment* depends on the most probable model chosen by the appliance recognition task: as a consequence, we could compare an *alignment* of a wrong model with the original labels. For this reason we compute two accuracy rates for the state recognition:

- The real accuracy rate (AR_R), when considering the *alignment* of the winning models.
- The oracle accuracy rate, (AR_{Or}), when considering the *alignment* of the correct models.

Given that AR_R is subject to misclassification errors of the appliance recognition, its value is expected to be lower than AR_{Or}. In Table 3.14 we report the averages of AR_R and AR_{Or} per classes when performing the state recognition for both protocols. As expected, the average AR_{Or} is greater than the average AR_R, even if the difference is small for the *Intersession* protocol. The states of some categories, as Mobile phone and Hi-Fi, seem more difficult to be correctly identified, providing often poor results. Others categories, as Computer, Lamp, Fan and Kettle, provide very good state recognition performances.

Comments on the results

For the *Intersession* protocol on the ACS-F1 database, the baseline accuracy rate is 94.2%, using the dynamic coefficients and with $K = 64$. As for GMMs, the inclusion of the velocity coefficients appears very beneficial, with an increment of more than 4%. The inclusion of the acceleration coefficients does not seem to improve the results. The weighted probability computation slightly improves the accuracy rate, yielding to an accuracy rate of 95.2%. The profile of the accuracy rate for the weighted

	Intersession		Unseen Appliance	
	AR_R (%)	AR_{Or} (%)	AR_R (%)	AR_{Or} (%)
Hi-Fi	81.9	76.8	69.7	70.2
Television	99.4	99.4	95.9	97.3
Mobile P.	66.6	64.8	67.5	70
Coffee M.	99.5	99.5	95.4	96.9
Computer	99.9	99.9	99.7	99.8
Fridge	84.8	84.7	87.8	91.2
Lamp Inc.	99.8	99.8	98.6	99
Laptop	97.3	97.9	96.4	96.5
Oven	99.8	99.8	99	98.7
Printer	91.1	91.2	60.3	71
Fan	99.9	99.9	99.7	97.7
Kettle	99.9	99.9	99.9	99
Lamp CFL	99.9	99.9	99.9	97.4
Monitor	93.5	99.9	99.5	99.8
Shaver	89.2	90.1	86.9	84.4
average	93.5	93.6	90.4	91.4

Table 3.14: The averages of AR_R and AR_{Or} for state recognition using both protocols.

probability computation is illustrated in Figure 3.21a. The best case is reported in Table 3.15 (left). As for the K-NN and GMMs, the Laptop is the category yielding the worst accuracy rate, while the other categories attain 89.0% of accuracy rate and several others obtain an accuracy rate of 100%. The highest misclassification is between Laptop - Computer.

For the *Intersession* protocol on the ACS-F2 database, the baseline accuracy rate is 95.8%. This result is obtained using the velocity coefficients and excluding the acceleration coefficients. Figure 3.21c illustrates the profile of the weighted probability computation. In the best case, it slightly improves the accuracy rate up to 95.2% compared to the baseline result when including all the dynamic coefficients. The confusion matrix of the best case is reported in Table 3.16 (left). As for GMMs, Mobile phone provides poor results, while all the categories attain at least 87%. As for K-NN and GMMs, the misclassification between Mobile phone - Shaver is important, while several categories obtain an accuracy rate of 100%.

For the *Unseen appliance* protocol on the ACS-F1 database, the best overall accuracy rate is 74.5%. As for the K-NN and GMMs, the inclusion of both dynamic coefficients is particularly beneficial. In this case the accuracy rate increases of 14.5%. In Figure 3.21b the profile of the weighted probability computation shows the importance of the dynamic coefficients. A balance between the original coefficients and the dynamic coefficients does not seem to sensibly improve the results. The confusion matrix is reported in Table 3.15 (right). Also in this case, the category Lamp provides poor performances (50%), however it is better than when using GMMs.

For the *Unseen appliance* protocol on the ACS-F2 database, the best overall accuracy rate is

	Intersession protocol										**Unseen appliance protocol**									
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp	Laptop	Oven	Printer
Hifi	.96	0	0	0	0	0	.01	.01	0	.02	.55	.05	.1	.05	0	.15	0	0	0	.1
Television	0	.92	0	.06	0	0	0	.01	.01	0	0	.6	0	0	.05	.15	.05	.05	0	.1
Mobile P.	0	0	.93	0	0	.06	0	0	.01	0	0	0	1	0	0	0	0	0	0	0
Coffee M.	0	0	0	1	0	0	0	0	0	0	0	0	0	.8	0	0	0	0	.1	.1
Computer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	.85	0	.05	.1	0	0
Fridge	0	0	0	0	0	1	0	0	0	0	0	.1	0	0	0	.8	.1	0	0	0
Lamp	0	0	0	.01	0	.04	.89	0	0	.06	.05	.05	.1	0	.15	.05	.5	.05	.05	0
Laptop	0	.06	0	0	.09	0	0	.82	.03	0	0	0	0	0	0	0	0	.9	0	.1
Oven	0	0	0	0	0	0	0	0	1	0	0	0	0	.3	0	0	0	0	.65	.05
Printer	0	0	0	0	0	0	0	0	0	1	.15	0	0	.05	0	0	0	0	0	.8

Table 3.15: In the left side, the confusion matrix of the Intersession protocol for HMMs with weighted probability computation. In the right side, the confusion matrix of the Unseen appliance protocol for HMMs with baseline test.

	Intersession protocol														
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver
Hifi	.99	0	0	0	0	0	0	0	0	.01	0	0	0	0	0
Television	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Mobile P.	0	0	.73	0	0	0	0	0	0	.03	0	0	0	0	.25
Coffee M.	0	0	0	.97	0	0	0	0	.03	.01	0	0	0	0	0
Computer	0	0	0	0	.99	0	0	.01	0	0	0	0	0	0	0
Fridge	0	0	0	0	0	.99	0	0	.01	0	0	0	0	0	0
Lamp Inc.	0	0	0	0	0	0	.99	0	0	0	.01	0	0	0	0
Laptop	0	.02	0	.01	0	0	0	.87	0	.07	.03	0	0	0	0
Oven	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Printer	0	0	0	.01	0	0	0	0	0	.99	0	0	0	0	0
Fan	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Kettle	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Lamp CFL	0	0	0	0	0	0	0	0	0	.03	0	0	.97	0	0
Monitor	0	0	0	0	0	0	0	.09	0	.01	0	0	0	.89	0
Shaver	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	.99

	Unseen appliance protocol														
	Hifi	Television	Mobile P.	Coffee M.	Computer	Fridge	Lamp Inc.	Laptop	Oven	Printer	Fan	Kettle	Lamp CFL	Monitor	Shaver
Hifi	.63	0	.07	.07	0	.1	0	.07	0	0	0	0	0	0	.07
Television	0	.53	0	.03	.17	.17	0	.03	0	0	0	0	0	.07	0
Mobile P.	0	0	.93	0	0	0	0	0	0	0	0	0	0	0	.07
Coffee M.	0	0	0	.77	0	0	0	0	.17	.07	0	0	0	0	0
Computer	0	0	0	0	.9	0	0	.1	0	0	0	0	0	0	0
Fridge	0	0	0	.07	0	.87	0	0	.03	0	.03	0	0	0	0
Lamp Inc.	.13	0	0	0	0	.07	.7	0	0	0	.1	0	0	0	0
Laptop	0	.03	0	0	.07	0	0	.77	0	0	0	0	.07	.07	0
Oven	0	0	0	.07	0	0	0	0	.87	.07	0	0	0	0	0
Printer	0	0	.07	.07	0	.07	0	0	0	.6	.07	0	0	0	.13
Fan	0	.07	0	0	0	.07	0	0	0	0	.87	0	0	0	0
Kettle	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Lamp CFL	0	0	.03	0	0	.07	0	0	0	0	0	0	.87	0	.03
Monitor	0	.07	0	0	0	0	0	.13	0	0	0	0	0	.8	0
Shaver	.13	0	.47	0	0	0	0	0	0	0	0	0	.07	0	.33

Table 3.16: In the left side, the confusion matrix for Intersession protocol on the ACS-F2 using HMMs with baseline test. In the right side, the confusion matrix for Unseen appliance protocol on the ACS-F2 using HMMs with weighted probability computation.

72.4%. As for the ACS-F2, the inclusion of the dynamic coefficients is particularly beneficial, with an improvement of 8%. In Figure 3.21d the weighted probability computation is illustrated. The tuning of α and β sensibly improves the results, up to 76.2%. The confusion matrix is given in Table 3.16 (right). As for the K-NN and GMMs, Kettle is the only category yielding 100% of accuracy rate. The Shaver category shows the worst performances with 33%. The larger misclassification errors are between categories Shaver - Mobile phone that appear very difficult to separate.

Discussion

Generally speaking, the results provided by HMMs are coherent between the two versions of the database and the protocols. Moreover, the results are in a certain way similar to those obtained using GMMs.

- When using the Intersession protocol, the inclusion of the velocity coefficients provides significant improvements, while the acceleration coefficients do not seem to be particularly effective. More specifically, for the ACS-F2, the inclusion of the acceleration coefficients deteriorates the accuracy rate.

- When using the Unseen appliance protocol, the inclusion of the dynamic coefficients provides significant improvements on both the databases. As for GMMs, a fewer number of Gaussians is needed for avoiding the overtraining.

- When using HMMs, the weighted probability computation is beneficial for all the databases and protocols. The trend of the profiles is very similar: when increasing the contribution of the dynamic coefficients, the accuracy rate increases up to a peak. However, if the original coefficients have a too small weight, the accuracy rate starts to decrease. For the Intersession protocol the difference between the starting value (using only the original coefficients) and the peak value is smaller than for the Unseen appliance protocol.

In Table 3.17 we report the best accuracy rates obtained for the three algorithms when using the ACS-F1 and ACS-F2 databases with the *Intersession* and *Unseen appliance* protocols. First of all, we notice that the generative modeling outperform the K-NN algorithm. For the *Intersession* protocol, we observe improvements of about 3-5%, while for the *Unseen appliance* protocol the improvements are around 7-10%. When comparing GMMs and HMMs, the differences appear very slight, around 0.5%. The only exception is represented by the *Unseen appliance* protocol on the ACS-F2, where we notice an improvement of about 3%.

	ACS-F1		ACS-F2	
	Intersession	Unseen appliance	Intersession	Unseen appliance
K-NN	92.0 $(k = 1)^{w}$	67.5 $(k = 9)^{w}$	90.2 $(k = 1)^{t}$	66.2 $(k = 1)^{w}$
GMMs	95.4 $(K = 40)^{w}$	75.0 $(K = 4)^{w}$	95.2 $(K = 40)^{w}$	73.1 $(K = 8)^{w}$
HMMs	95.2 $(K = 68)^{w}$	74.5 $(K = 4)^{b}$	95.8 $(K = 76)^{b}$	76.2 $(K = 8)^{w}$

[b] Baseline
[t] Thresholding
[w] Weighted computation

Table 3.17: Best accuracy rates for the three machine learning techniques under analysis: K-NN, GMMs and HMMs.

3.6.4 Evaluation of the features

In spite of the known correlation existing among the features provided by the database (as per the laws of electricity), we want to evaluate, from a machine learning perspective, the information brought by features and coefficients in the context of the identification task. Generally speaking, the features used for a classification task have to be *relevant* and *not redundant*. A feature is considered *relevant* if it is highly correlated to the classes and *redundant* if it is highly correlated to other features [83]. Such information can be provided through the analysis of the correlation between variables. Two approaches are mainly used: computing the *linear correlation* or using the *information theory*. When the variables are not linearly correlated, the first approach is scarcely effective. In this work we use the second approach. We compute the *entropy* of the features [83].
The entropy H of a feature X can be computed as follows:

$$H(X) = -\sum_{i} p(x_i) log_2(p(x_i)) \tag{3.9}$$

where $p(x_i)$ are the prior probabilities of X. The entropy of the feature X when observing another feature Y is defined as:

$$H(X|Y) = -\sum_{j} p(y_j) \sum_{i} p(x_i|y_j) log_2(p(x_i|y_j)) \tag{3.10}$$

where $p(x_i|(y_j))$ are the posterior probabilities of X given the values of Y. Using the previous two equations, we can compute the information gain (IG) as:

$$IG(X|Y) = H(X) - H(X|Y) \tag{3.11}$$

The *symmetrical uncertainty* (SU) is chosen for compensating the IG bias and for normalizing the values:

$$SU(X, Y) = 2\left[\frac{IG(X|Y)}{H(X) + H(Y)}\right] \tag{3.12}$$

In Figure 3.22 the SU is depicted using a gray scale. For finding the "not relevant" features we check the attributes for which SU is lower than a prefixed threshold. In our case, the network frequency and its dynamic coefficients (lines 2, 8, 14 in Figure 3.22) have been categorized as not relevant. As expected, the small deviations on the network frequency are substantially independent to the appliance category.

As second step we determine the redundant features. We use the information available from the theory of electricity instead of the SU matrix for sake of simplicity. The phase angle, reactive power and active power are redundant as the third one can be computed from the other two. The phase angle has the lowest SU value among the three, therefore this feature and its dynamic coefficients are selected as redundants. Following the same procedure we analyze the RMS voltage, the RMS current and the active power. We select the RMS voltage and its dynamic coefficients as redundants because it has the lowest SU value among the three features. From the electrical point of view, removing the network frequency and the RMS voltage is an operation that totally makes sense, given that these two features are imposed by the network and stable for their own nature.

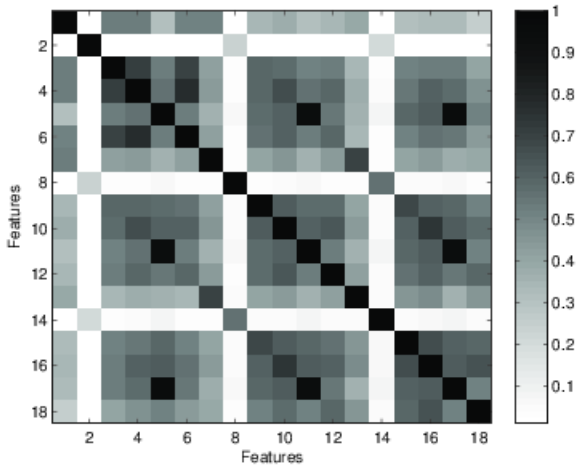


Figure 3.22: Symmetrical uncertainty in gray scale. From the position 1 to 6 there are the original features with respectively phase angle, network frequency, reactive power, real power, RMS voltage, RMS current. From 7 to 12 there are the velocity coefficients and from 13 to 18 the acceleration coefficients.

As next step, we evaluate the impact of the feature space reduction through the evaluation of the classification results. We observe the accuracy rate when using only one feature and its coefficients. In Figure 3.23 we illustrate the accuracy rates for both databases and protocols. The observations previously reported are confirmed:

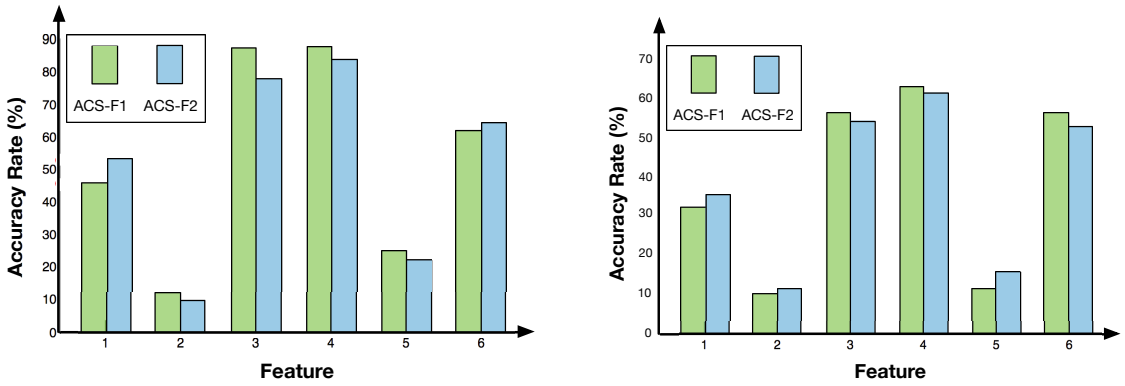
- *network frequency* is a "not relevant" feature, yielding to an accuracy rate slightly better than random (10% for the ACS-F1 and 6.6% for the ACS-F2).
- *RMS voltage* yields to low accuracy rate when using the Intersession protocol and it could be considered as "not relevant" feature for the Unseen appliance protocol: it can not be used for generalizing the appliance category.
- *Phase angle* and *RMS current* appear useful for the classification task, however for all protocol and databases the *RMS current* performs better than the *Phase angle*, especially for the Unseen appliance protocol.
- *Active Power* and *Reactive Power* are fundamental for the classification task, with a slightly preference of the *Active Power* on the *Reactive Power*.

As last step, we provide the accuracy rate after the feature selection procedure. We perform the tests by increasing the number of Gaussians and using the selected features with the relative dynamic coefficients. As shown in Table 3.18 the accuracy rates are generally better than those obtained without the feature selection. For the K-NN we also perform the thresholding, given that it seems to be particularly effective for this algorithm. As we can observe, the K-NN greatly improves the accuracy rates, approximately between 5% and 7%. For GMMs and HMMs the improvement concerns principally the Intersession protocol. Increments approximately between 1-3.5% are estimated for the Intersession protocols, while approximately the 1% for the Unseen appliance protocol.

3.7 Performances of generative modeling

In this work, we opt to use generative models as GMMs and HMMs for performing appliance classification. Two main reasons are behind this choice: (i) the scalability and (ii) the adaptability.

Generally speaking, generative models are more scalable than discriminative models. In generative modeling the training of each model is separated, because each model uses only its specific data. This property is very useful when adding a new class to an already trained system. In generative modeling



(a) Intersession protocol

(b) Unseen appliance protocol

Figure 3.23: Feature selection using GMMs. For sake of simplicity, we report the features as numbers, listed in the order: phase angle, network frequency, reactive power, real power, RMS voltage, RMS current.

	ACS-F1		ACS-F2	
	Intersession	Unseen appliance	Intersession	Unseen appliance
K-NN	90.0 ($k = 1$)	54.0 ($k = 9$)	90.2 ($k = 1$)	57.3 ($k = 1$)
K-NN$_{fs}$	96.0 ($k = 2$)	61.0 ($k = 6$)	96.0 ($k = 1$)	62.2 ($k = 8$)
GMMs	94.5 ($K = 40$)	73.5 ($K = 4$)	94.0 ($K = 32$)	72.2 ($K = 8$)
GMMs$_{fs}$	96.5 ($K = 40$)	74.5 ($K = 12$)	95.5 ($K = 40$)	73.3 ($K = 8$)
HMMs	94.2 ($K = 68$)	74.5 ($K = 4$)	95.0 ($K = 72$)	72.4 ($K = 8$)
HMMs$_{fs}$	97.9 ($K = 76$)	75.5 ($K = 8$)	96.4 ($K = 76$)	73.3 ($K = 8$)

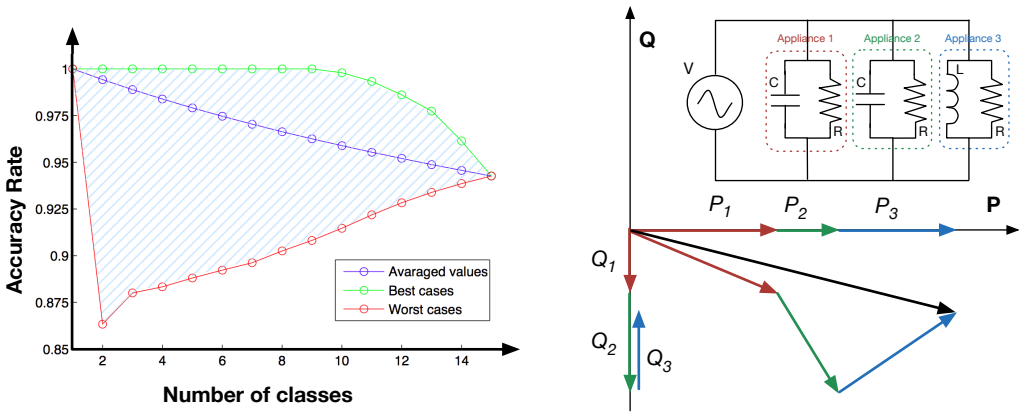
Table 3.18: Accuracy rates before and after the feature selection. For the K-NN we perform the thresholding, while for the GMMs and HMMs we report the baseline analysis.

only the new model has to be trained and simply added to the existing system. On the opposite case, in discriminative modeling data of all models is used at one time for training and when adding a new class, all the system has to be trained again.

Generative models are more adaptable to a specific environment than discriminative. For instance in a realistic scenario some appliances can have a low probability to appear, depending on the activity context, e.g. using a Shaver at the working place, or the environmental context, e.g. using a Fan during winter. By using such information, the priors in the Bayes' law can be modified on the fly. The priors can also be increased using the contextual information, e.g. considering that it is very likely to find Laptop, Computer and Monitor at the working place. As stated in the previous chapter, discriminative models directly compute the a posteriori probability, without giving the opportunity to easily modify the priors on the fly.

3.7.1 Classification with label selection

In order to show the scalability/adaptability of our system, we compute the accuracy rate when reducing the number of classes. The accuracy rate is computed on the ACS-F2 database using the Intersession protocol. As machine learning algorithm we use a GMM with 40 Gaussians and we apply the feature selection. The number of classes has been varied between 1 and 15. Depending on the chosen classes, the accuracy rate is different. For this reason we report in Figure 3.24a three plots: in blue the average curve, in green the best case and in red the worst case. The average curve is the average of all the possible combinations of the classes. The number of combinations depends on the



(a) (b)

Figure 3.24: a) Accuracy rate when varying the number of classes and b) example of a circuit with more than one appliance in the P-Q plane.

number of classes and is computed with the binomial coefficient:

$$\binom{n}{k} = \frac{(n)!}{(n-k)!\,k!} \tag{3.13}$$

where n is the total number of classes in the ACS-F2, therefore 15, and k is the number of chosen classes, therefore it varies between 1 and 15. The average curve has a behavior almost linear and the smooth trend of the curve is due to the average operation. The green curve is the best case among all the combinations. If we randomly pick k classes, it would be the luckiest guess. Up to 9 classes we can achieve 100% of accuracy rate if choosing the "right" combination. The red curve is the worst case among all the combinations. If we randomly pick k classes, it would be the unluckiest guess. With the exception of the degenerated case $k = 1$, the red curve appears to grow. The worst case is when the two hardest classes to separate are chosen (i.e. Mobile phone and Shaver) and adding more classes improves the accuracy rate. This pair of categories is also the one yielding to the biggest misclassification error, as represented in Table 3.11 (right). In blue we represent the area of the possible accuracy rates independently of the number of chosen classes and chosen combinations.

In this case we suppose the priors of the categories as uniformly distributed, i.e. equal to 0 if absent and equal to 1 if present. The priors are after normalized for giving the sum equal to 1. However, in some specific applications a different value can be given to the priors, for instance by computing the statistical distribution of the categories and proportionally computing the priors.

3.7.2 Multi-signal detection

The adaptability of the generative modeling is evident also when performing the multi-signal detection. Up to now we have analyzed signals containing only one signature at a time. However, if we have signatures aggregated together, we can try to detect which appliances give their contribution. By using the laws of the electricity and knowing that appliances are plugged in parallel on the network, we could use the adaptive property of some features, as the active and reactive power. In Figure 3.24b we illustrate an example of a circuit with two capacitives and one inductive appliances. Their behavior is also depicted in the P-Q plane where the complex powers are simply added as vectors. To add or subtract two or more vectors, the corresponding components, in this case the active powers and reactive powers, have to be added or subtracted.

For understanding which appliances contribute to an aggregated signal, one possibility is to model each combination and test the aggregated signal against all the possible combinations. In order to build these models, two solutions are available. The first consists in training the models with examples derived from combinations of single signatures. This procedure can be very heavy and time-consuming, given that the number of combinations exponentially grows with the number of available classes. The

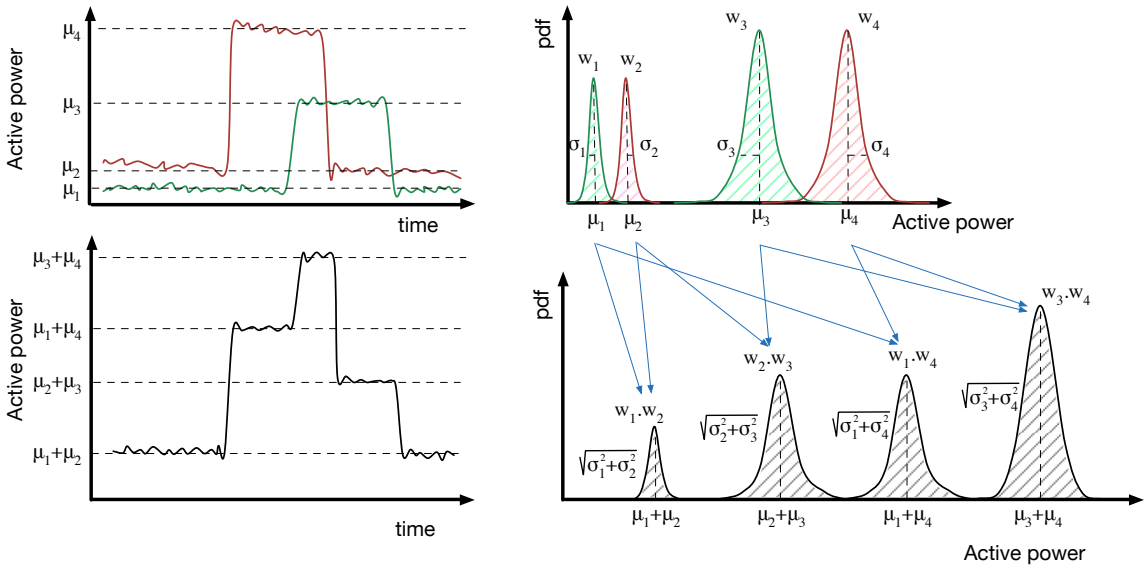


Figure 3.25: Example of the model merging for the aggregated signature identification.

second solution consists in merging the model computed for the single classes, without the need to use the combined signatures. The first solution can be adopted by discriminative and generative models, while the second only by generative models.

However, how exactly these models can be merged? We show an example of such a system by using three features: the active power, the reactive power and the current, without computing the dynamic coefficients. When two or more signatures are aggregated, the final active power is determined by the sum of the active power of all the signatures (as shown on the Figure 3.24b). The same happens for the other two features, the reactive power and the current. When using GMMs, every category is represented by a given number of Gaussians. For computing the combination of two or more models, we can use an interesting property of the Gaussians: if X and Y are independent random variables normally distributed, then their sum is also normally distributed [84]. The sum of two independent normally distributed random variables is normal, with its mean being the sum of the two means, and its variance being the sum of the two variances:

$$\begin{cases} X \sim \mathcal{N}(\mu_X, \sigma_X^2) \\ Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2) \\ Z = X + Y \end{cases} \tag{3.14}$$

then:

$$Z \sim \mathcal{N}(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2) \tag{3.15}$$

We illustrate this property by an example on Figure 3.25. We assume two different signatures, one represented in green and the other in red. In the upper-right side of the Figure we represent the probability density functions when using the GMMs algorithm with two Gaussians on each model. The first model, in green, has two mixtures (μ_1, σ_1^2, w_1) and (μ_3, σ_3^2, w_3). The second model, in red, has also two mixtures (μ_2, σ_2^2, w_2) and (μ_4, σ_4^2, w_4). Our aim is to find the model representing the combination of the two. The method of model merging consists in using the property of the sum of two independent normally distributed random variables. Starting with the two models, we can combine them in order to obtain the model representing the combination of the two. Every mixture of every model has to be added to every mixture of the other model. The weight of the resulting mixture is the product of the weights of the original mixtures. The combined model therefore has four mixtures ($\mu_1 + \mu_2$, $\sigma_1^2 + \sigma_2^2$, $w_1 \cdot w_2$), ($\mu_2 + \mu_3$, $\sigma_2^2 + \sigma_3^2$, $w_2 \cdot w_3$), ($\mu_1 + \mu_4$, $\sigma_1^2 + \sigma_4^2$, $w_1 \cdot w_4$) and ($\mu_3 + \mu_4$, $\sigma_3^2 + \sigma_4^2$, $w_3 \cdot w_4$).

By using the Intersession protocol and a GMM with $K = 40$ on the ACS-F2, we attain 90.9% of accuracy rate. We choose $K = 40$ given that is the best number of Gaussians when working with a

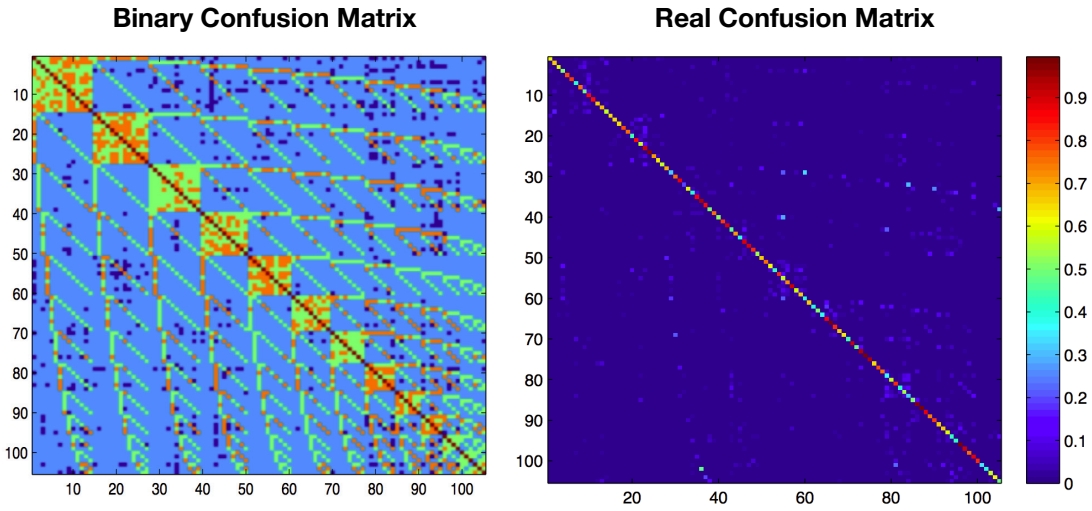


Figure 3.26: Confusion matrices of the merged GMMs system in the task of detecting two aggregated signatures. The *Binary Confusion Matrix* is used for better understanding the differences between the zones of the confusion matrix. The red points indicate when both signatures are correctly identified. In orange and green there is the zone of the matrix where one over two signatures is correctly identified. In bright blue and dark blue there is the zone of the matrix where any signature is correctly identified. The *Real Confusion Matrix* contains the accuracy rates represented in a scale of colors.

reduced set of features, as shown in Table 3.18. As previously described, we build 15 different models using the data in the training set and the performances are computed using the signatures in the test set. We find all the combinations of all the models by using the Formula 3.13: when adding two signatures belonging to two different categories, we have 105 combined models. By using the property previously described, we compute the Gaussians for every combined model.

As next step, we merge all the signatures in the test set in order to form a test set of combined signatures for measuring the performances of our system. The number of all the possible combinations of signatures belonging to different categories can be computed through the following Formula:

$$c_s = \binom{n}{k} m^k = \frac{(n)!\ m^k}{(n-k)!\ k!} \tag{3.16}$$

where $n = 15$ is the number of categories, $m = 15$ is the number of appliances per categories and $k = 2$ is the number of signatures that are combined together. Under these conditions, the number of all the possible combinations of signatures c_s is equal to 23625. In 67.0% of the cases we identify both appliances, in 28.7% we identify one appliance on two and in 4.4% we do not identify any of the signatures. In Figure 3.26a we report the confusion matrix of all the 105 models. In order to simplify the interpretation of the confusion matrix, we plot two different confusion matrices. The first, called *Binary Confusion Matrix* is used for better understanding the differences between the zones of the confusion matrix. The red points (on the diagonal) indicate when both signatures are correctly identified. In orange and green there is the zone of the matrix where potentially one over two signatures is correctly identified. If the point is green, therefore no error has been made, otherwise, if the point is orange, at least one signature has been misclassifed. In bright blue and dark blue there is the zone of the matrix where no signature is correctly identified. If the point is bright blue, therefore no error has been made, otherwise, if the point is dark blue, at least one signature has been misclassifed. In Figure 3.26b we report the *Real Confusion Matrix*, containing the accuracy rates represented in a scale of colors. As expected, the diagonal contains the best accuracy rates, however the structure represented by the orange/green points of the *Binary Confusion Matrix* is sometimes visible.

3.8 Applications

3.8.1 Real-time application

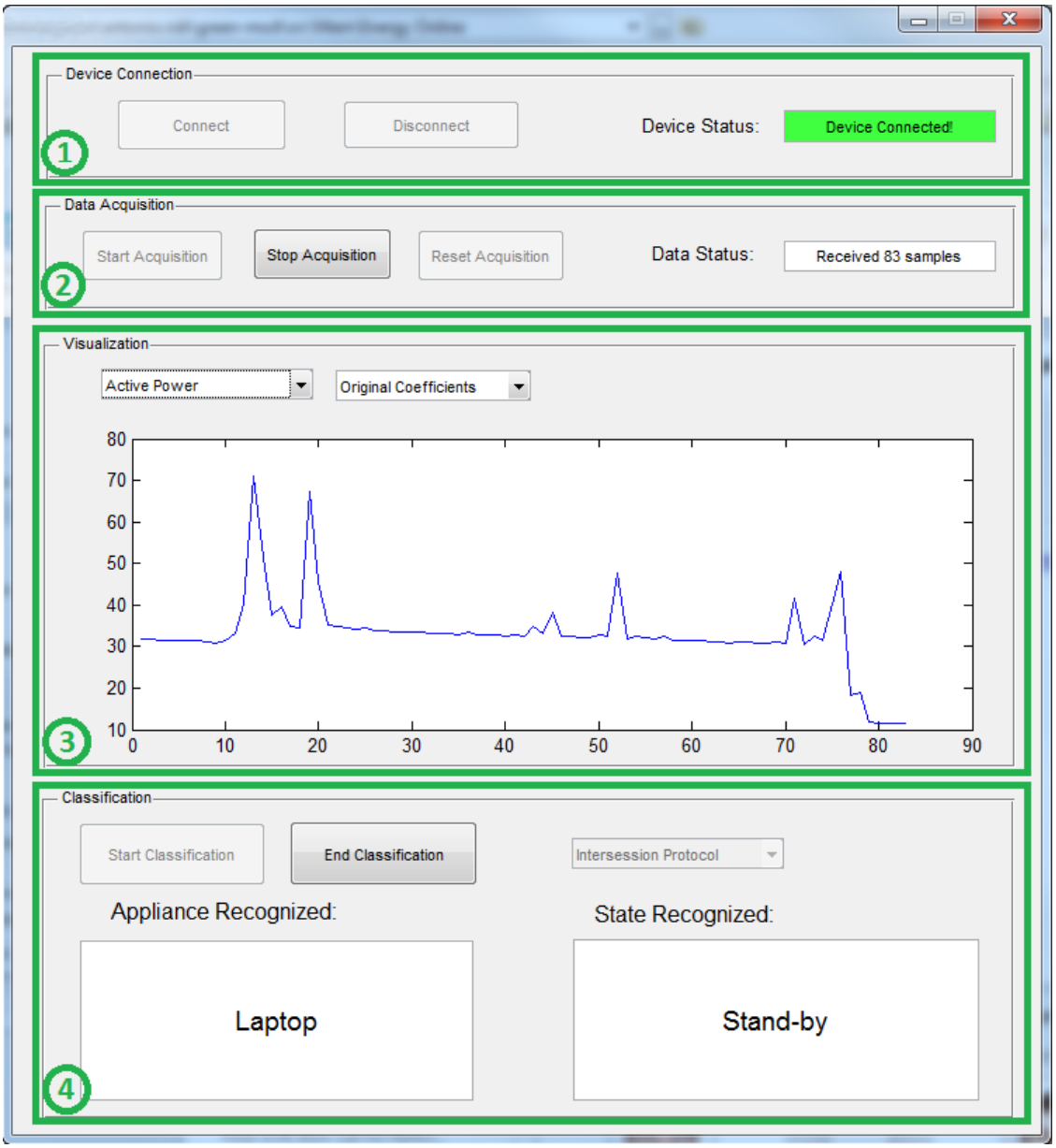
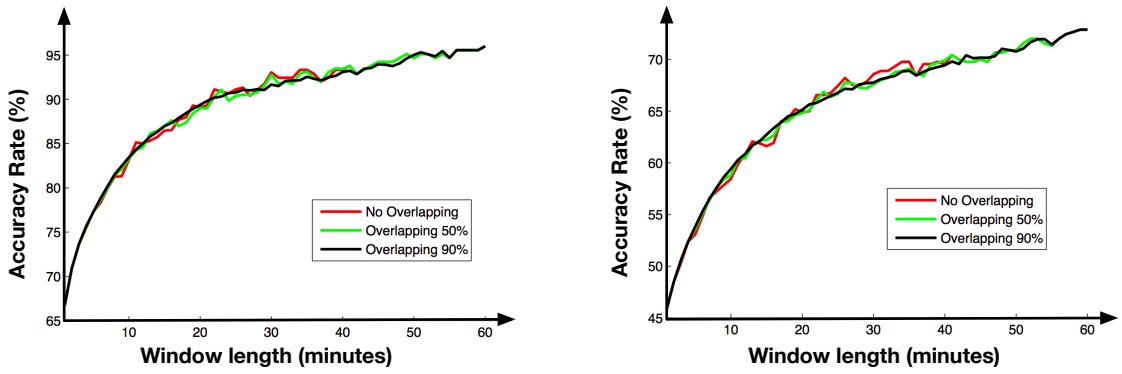


Figure 3.27: Interface of the real-time application for the appliance and state recognition. The application has four parts. Part 1 gives the status of the device connection. Part 2 describes the status of the communication and the number of received samples. In part 3 the received data are visualized and the plot is systematically updated when a new samples are received. Finally in part 4 the appliance and state recognized are presented.

In the previous section we have shown off-line system performances for a classification task aiming at identifying the category and the state of appliances. Here we want to show the feasibility of a system working in real-time on data acquired from a plug by using the HMMs parameters previously learnt.

We realize a system able to provide in real-time the same features of those contained in the ACS-F2 database. The data is provided with the same sampling frequency, i.e. every 10 seconds we receive



(a) Intersession protocol

(b) Unseen appliance protocol

Figure 3.28: Trends of the accuracy rates for the *Dynamic Intersession* (a) and *Dynamic Unseen appliance* (b) protocols, no overlapping in the windowing operation (red), overlapping of 50% (green) and overlapping of 90% (black).

new data. The samples are pre-processed as previously described. The samples are progressively stored for being used for the recognition task. We use the emission and transition matrices of all the models for the best case using the *Intersession* protocol. The interface of our application is presented in Figure 3.27. Four parts have been implemented:

1. **Device connection**. The application establishes the connection with the smart plug and gives a positive or negative feedback to the user depending on the connection result.

2. **Data communication**. When the device is connected, the communication can be started, stopped or reset by the user. The feedback on the total number of samples received is provided to the user.

3. **Data visualization**. The original features are shown in real-time. A popup menu allows to select the desired features. After 50 seconds from the start of the communication (the size of the analysis window) the dynamic coefficients can be computed and also visualized in real-time.

4. **Appliance and state identification**. After the computation of the dynamic coefficients, this last layer can be activated. We use the Viterbi algorithm to determine the *alignment* and *likelihood* of all stored samples. Two classifications are effectuated in real-time, consisting on the identification of the appliance category and the state of the identified appliance. The actual state corresponds to the last state of the *alignment*. The results of the classifications are shown in the bottom part of the interface.

The system provides the first classification results after less then one minute from the start of the data communication. For obtaining similar accuracy rates of those reported in the previous section we should ideally wait the same time duration as the one provided in the training instances, namely one hour. Using shorter time duration the results are expected to drop. By using the *Variable Time Length* protocol we can determine the variation of the accuracy rate with the time duration.

Variable Time Length

We report here on the trend of the accuracy rates for the *Dynamic Intersession* and the *Dynamic Unseen appliance* protocols. As stated above, we use the whole duration of the signature for the training phase, i.e. 1 hour. For the test phase we use windows of different lengths: we start with windows of 1 minute length and we iteratively increment the length up to 60 minutes by steps of 1 minute. A total of 60 accuracy rates are computed. We apply the feature selection for both protocols as explained in [85] and in Section 3.6.4.

Figure 3.28 shows the trends of the accuracy rates for the *Dynamic Intersession* and the *Dynamic Unseen appliance* protocol on the ACS-F2 database. In red we report the trend with no overlapping in the windowing operation, in green with an overlapping of 50% and finally in black with an overlapping of 90%. We observe that larger overlapping leads to smoother performance curves. However, the performances are not so much different. As a backdrop, an increment of the overlapping causes a higher number of windows to be tested and, as a consequence, a higher computational time. As expected, for both protocols the accuracy rate increases as a function of the window length. We observe a steep increase of the performance from 1 to 20 minutes while the gain is more linear from 30 to 60 minutes.

As we want other researchers to be able to precisely compare their results with ours, we provide the details of a model fitting the trends of the accuracy rates. The model used for the fitting is presented by the following Formula:

$$y = a\ exp^{(bx)} + c\ exp^{(dx)} \tag{3.17}$$

where a, b, c and d are the fitting coefficients. In Table 3.19 we report the coefficients of the fitted curve that best approximates the trend of the accuracy rates obtained with an overlapping of 90%.

Protocol	**a**	**b**	**c**	**d**
Dynamic Intersession	87.85	1.47e-3	-23.51	-0.1451
Dynamic Unseen appliance	64.04	2.12e-3	-20.38	-0.1244

Table 3.19: Coefficients of the fitted curves as expressed in the Formula 3.17.

3.8.2 Virtual sensors for virtual networks

We evaluate our machine learning algorithms on a concrete scenario of appliance recognition in a real context, leveraging on the Web-of-Things technologies to facilitating integration into networks. As described in Section 3.2.1, the weakness of a centralized architecture is in the server that is a single point of failure. In our works [86, 87], we show a practical application where machine learning algorithms are distributed on the sensor network for the appliance recognition. Such system has the benefit of robustness with a distribution of the tasks required to perform an identification. We also leverage on the "natural distributivity" of the computation which is allowed by generative models. In fact, when analyzing the distribution capabilities of machine learning algorithms, we have two different approaches depending if the algorithm is generative and discriminative. Both approaches have the possibility to distribute the computation, however generative models naturally offer the possibility to compute each model distribution independently of the others. In practice, every node is able to manage a separate computation of class conditional probabilities. On the other hand, discriminative models as ANNs or SVMs share class parameters making much more difficult the distribution of the computational load.

As illustrated in Figure 3.29, we introduce two entities able to manage this distributed architecture: (i) *Virtual Sensor* and (ii) *Virtual Class*. The *Virtual Sensor* is a non-physical sensor that appears as a regular one within the network. The objective is to hide the complexity of the computational data management and make the detection task appear in the network as a conventional sensor. In this way, the integration in a multi-sensor network can be easily done eventually using IoT paradigms. The machine learning algorithms are distributed among the *Virtual Classes*. They store the mathematical models able to perform the machine learning task. *Virtual Sensor* manages the information coming from several *Virtual Classes*. Focusing on generative modeling, suppose to have a set of possible classes $C = \{c_1, c_2, \ldots, c_M\}$, where M is the total number of classes and an input vector $\mathbf{x}$. Referring to the Bayes' law, we want to compute the posterior probability $p(c_m \mid \mathbf{x})$ given $p(x \mid c_m)$, i.e. the likelihood of c_m with respect of $\mathbf{x}$, and $p(c_m)$, i.e. the prior probability of the class c_m. As previously said, generative modeling has a "natural distributivity" of the computation. In fact, the computation of the different likelihoods are independent the one from the other. Each *Virtual Class* contains a generative model able to compute the likelihood of the input vector $\mathbf{x}$. The *Virtual Sensor*

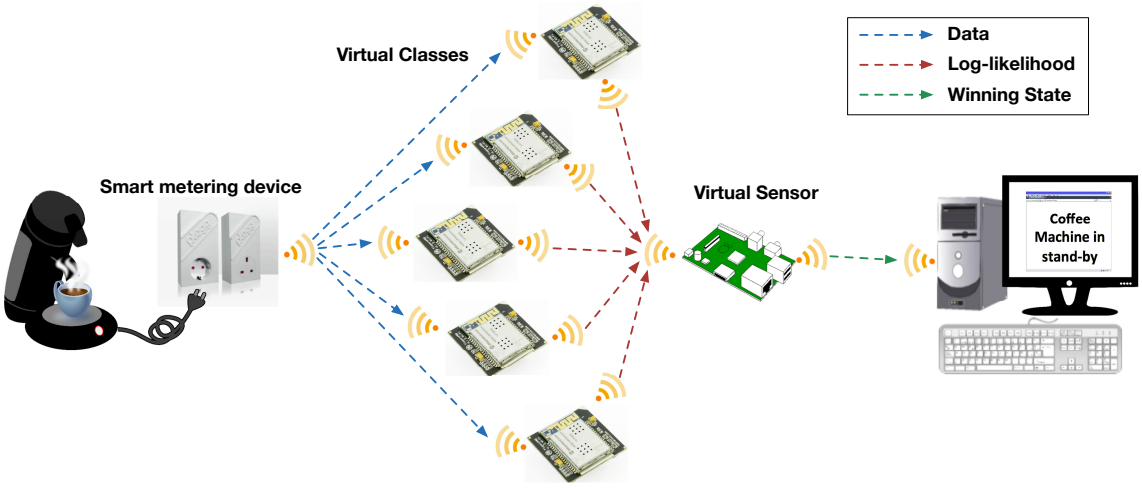


Figure 3.29: The real-time system able to perform the appliance recognition through the computational distribution using generative modeling [87]. An electrical signature of a device, in the example a coffee machine, is recorded and sent to devices, called *Virtual Classes*, storing different models previously trained. Each *Virtual Class* computes the log-likelihood that the signature belongs to the model and send this information to the *Virtual Sensor*. The *Virtual Sensor* uses the likelihoods and the priors about the classes for electing the winning model. This information along with the actual state is sent on the network.

manages the likelihoods coming from the *Virtual Classes*. It compares the likelihoods by using the information about the prior probability of the different *Virtual Classes*. Finally, the *Virtual Sensor* is able to communicate to the network the winning *Virtual Class*, the one providing the higher posterior probability $p(c_m \mid \mathbf{x})$ for the input vector $\mathbf{x}$.

In our implementation [87], we realize a real-time system by using a Plogg as smart metering device, some OpenPicus Flyport PRO WiFi[4] acting as *Virtual Classes* and a Rasperry Pi[5] acting as a *Virtual Sensor*. The information processed by the *Virtual Sensor* is sent to an interface device able to visualize this information. The Plogg device measures a sample every 10 seconds and the likelihoods are computed with the same frequency. Every time a new likelihood is computed, the *Virtual Classes* notify the *Virtual Sensor* and it takes a new decision. This architecture avoids the sequential requests of *Virtual Classes*, moreover the scalability and the round-trip time are improved. From the client point of view, the notification of the *Virtual Sensor* is observed when the appliance category changes.

We choose HMMs as generative algorithm for the recognition of the category of the appliance signatures. We train the HMMs offline due to the large quantity of data requested by the machine learning training procedures. For the training, we use five different classes coming from the ACS-F2 database, namely Coffee machine, Laptop, Lamp Inc., Lamp CFL and Mobile phone. Given that the evaluation of the performances will be done following the *Intersession* protocol, only the signatures contained in *session 1* will be used in the training phase. Due to the memory limitation of the Openpicus, we have to limit the number of Gaussians per class to 16. Through the Viterbi algorithm, each *Virtual Class* is able to retrieve the log-likelihood of an upcoming signature to the stored model, represented by matrices of means, standard deviations and weights. The *Virtual Sensor* compares the log-likelihood coming from the *Virtual Classes* and decides the winning category. In addition to the log-likelihood, the *Virtual Classes* systematically notify the information about the actual state, i.e. the last of the sequence of states computed by the Viterbi algorithm. In this manner, the *Virtual Sensor* can notify to the network the actual state of the signature by choosing the one corresponding to the winning *Virtual Class*. In order to evaluate the performances of our implementation, we use the signatures of the five classes in the ACS-F2 database in the *session 2*. In practice, we simulate a smart meter device by using the signatures of the database. We obtain an accuracy rate of 90%, a

[4]http://space.openpicus.com/u/ftp/datasheet/datasheet_flyportpro_wifi.pdf
[5]https://www.raspberrypi.org/

fair result considering the limited number of Gaussians that we have to use.

3.9 User Interaction Recognition

Our system is able to recognize the appliance class, such as Coffee Machine, Printer or Kettle. It is also able to recognize the sequence of states, such as active, non-active or stand-by, by using HMMs. In this additional work, we add a new layer to our system architecture called User Interaction Layer. As illustrated in Figure 3.30, this layer aims at inferring the moments during which the user interacts with the appliance. These moments are called User Interaction events. This layer uses as input the information coming from HMMs, i.e. the recognized appliance class and the sequence of states. The User Interaction events are derived from the analysis of the transitions in the sequences of states and a ruled-based system adds or removes these events depending on the recognized class.

3.9.1 Categories Definition

For the majority of the appliances, User Interaction events can be recovered through an analysis of the sequence of states. Usually, an interaction with a device implies a human triggering a command that will bring a transition of state of the appliance. HMMs can potentially retrieve this information from the hidden state transitions.

The difficulty in retrieving the user interaction events depends on the category the device belongs to. Several categorizations are possible. Hart [10] proposes a categorization depending on the operational state: two-state devices, multi-state appliances and continuously variable devices. Lee et al. [41] propose another separation based on the temporal usage of home appliances: *Background Appliances*, *Occupancy-Reactive Appliances* and *User-Interactive Appliances*. Here we refer to the following separation proposed by Zaidi et al. [52] that we have extended:

- **Usage dependent appliances** (UDA). The devices change their state depending on the interaction with the user (e.g. Microwave). A User Interaction usually then corresponds to a change of state.

- **Fixed operation appliances** (FOA). The device is characterized by a pre-established sequence of operations that are started by the user. If every fixed operation is represented by a hidden state, the User Interaction will imply a change from a specific state to another but will not involve the other transitions. For instance, a washing machine could be switched off (state 1), wash (state 2), rinse (state 3), spin (state 4) and maintenance wash (state 5). In this case the transition from state 1 to 2 implies User Interaction, but not the other transitions. In some cases FOA could be modeled by grouping a certain number of hidden states in one, for instance the washing machine could be "off" (state 1) or "on" (state 2).

- **Thermostatically controlled appliances** (TCA). These are appliances that either provide heating or cooling depending on the temperature of a certain environment. In this case the electrical consumption is not directly related to any user activity. Usually Fridges are considered an example of TCA.

- **Battery based appliances** (BBA). Finally we have to consider that some appliances are battery based. Their energy consumption is typically measured during the charging phase.

Several types of appliance could also be considered as composed by two or more systems. For instance, the Fridge usually has a lamp that automatically switches on when the door opens, making the device belonging to both TCA and UDA. Another example is represented by Laptop, that changes the energy consumption depending on their usage and the battery charge level. In this case Laptop can be considered as belonging to both BBA and UDA.

3.9.2 User Interaction Layer

The system described in the previous section is able to detect the appliance class and the state sequence without explaining the User Interaction with the appliances. For this reason we create an additional

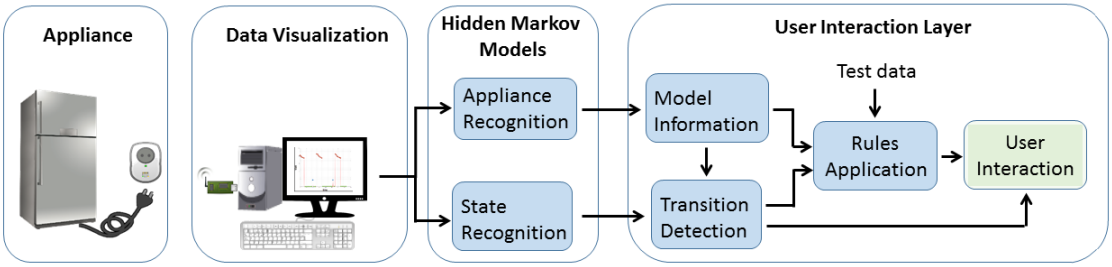


Figure 3.30: The system architecture: the appliance, recorded with a Smart Plug, communicates with a PC on which a HMM-based algorithm runs. The output of HMMs (i.e. the appliance class detected and the sequence of states) is taken as input of the *User Interaction Layer* that finds the User Interaction events.

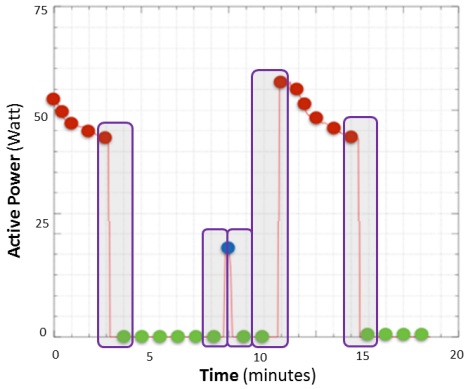


Figure 3.31: The time-series of the active power of a Fridge is represented. We use points of different colors according to their state: green for a non-compression phase, red for the compression phase and blue for the door opening. The state transitions are highlighted by using rectangular boxes.

layer, called *User Interaction Layer*, that takes as input the output of HMMs and which is able to infer the User Interaction events. In Figure 3.30 we represent the new system architecture: the output of HMMs is taken as input of the *User Interaction Layer*.

In most cases User Interaction events involve a change of the operational appliance mode and therefore a change of its state. Depending on the modeling choice, this change could correspond to a change of the model hidden state. As a consequence, the analysis of the transitions of the state sequence could provide information on the User Interaction events. Not all the transitions recovered by the state sequence are User Interaction events, but depend on the appliance class and its category (UDA, FOA, TCA or BBA). The appliance class is available from the output of HMMs which is bound to the category to which it belongs to. In Figure 3.31 we show an example using the transition between states. The time-series of the active power of a Fridge is represented using points of different colors depending on their state, i.e. green (state 1) for a non-compression phase, red (state 2) for the compression phase and blue (state 3) for the door opening. Some of the state transitions are indicators of User Interaction events, while others are not. However, knowing that the appliance is Fridge allows to be aware that the transitions involving the door opening (state 3) are due to User Interaction events.

By using the information coming from the state transitions and the winning model, we are able to determine if a user has interacted with a device for most cases. For some other cases, the system shows some limitations: (i) the user interaction happens inside the same state (*interaction without transition*) and (ii) under specific conditions the transitions have different meanings and have to be filtered (*transition selection*).

- **Interaction without transition**. This circumstance occurs when we choose to model a device

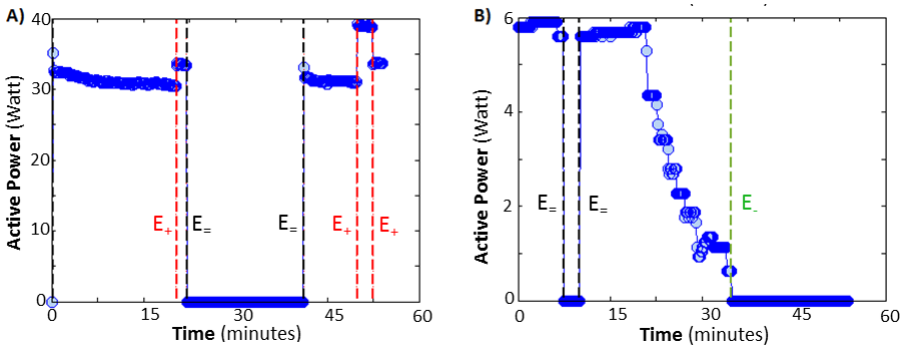


Figure 3.32: A) Fan signature having 3 power levels and B) Mobile phone signature. In black the events detected by the analysis of the transitions ($E_{=}$). In red the events added (E_{+}) and in green (E_{-}) the one removed by the rule-based algorithm.

with a lower number of hidden states than necessary, therefore a single state of the model could represent more than one real state. As a consequence, state transitions do not detect all User Interaction events. To detect the missed interactions, we apply certain rules on portions of the signal having a stationary hidden state. For instance, the Fan class of the ACS-F2 database includes devices of different brands and/or models with a different number of real states, for example mechanical fan with 2, 3 or more power levels. Given the heterogeneity of the devices, we have decided to represent the Fan class with a 2-state HMM: "active" and "not active" states. If a specific fan has more than 2 power levels, these are grouped inside the "active" state. As a consequence, when the user increases or decreases the fan speed without passing through the "not active" state, the interaction is not automatically detected. A Fan signal is illustrated in Figure 3.32a. This problem is solved by analyzing the portions of the signal having an "active" hidden state. With a simple rule-based system, we are able to recover the User Interactions events that do not have a transition in this portion of the signal.

- **Transition selection**. This circumstance occurs when the transition from one state to another has different meaning depending on the context: very often the transition is caused by a User Interaction event, however not in all cases. For the selection of the transitions related to User Interactions we apply a rule-based algorithm on the portions of the signal corresponding to the transitions. For instance, for the Mobile phone class we use a model with two hidden states: one for the charging phase ("on") and the other for the charged phase ("off"). The transition from "off" to "on" states means a User Interaction event (typically the user plugs the device or is using it). The transition from "on" to "off" could have two reasons: the device is charged or the device has been unplugged. In the first case, no user interaction event should be notified, while the opposite should occur in the latter case. Using a rule-based system, we verify if some conditions are satisfied: if the difference of the active power between the two states exceeds a threshold, the device has probably been unplugged (User Interaction), in the opposite case the device is fully charged (no User Interaction event). A Mobile phone signature is illustrated in Figure 3.32b.

In summary, for solving the system limitations, i.e. *interaction without transition* and *transition selection*, we apply a simple rule-based system. In both cases the rules are determined by using decision trees with binary splits for classification. The decision trees are built by using the examples in the training data. The decision trees can be translated in a sequence of conditions (rules), for instance for the Printer class we obtain the rules depicted in the Algorithm 6. The parameter ΔP is the velocity coefficient of the active power, ΔQ is the velocity coefficient of the reactive power and *class* determines if the event should be considered or not.

```
if ΔP < 8.7225 then
    if ΔQ < −5.4495 then
        class = 1
    else
        class = 0
    end
else
    class = 1
end
```

Algorithm 6: An example of a rule for the Printer class, obtained with decision trees.

3.9.3 Test Methodology

We perform some tests on the data of the ACS-F2 database in order to determine the accuracy of the recognition of the User Interaction events. We recover the ground truth by manually finding the User Interaction events in each signature. This task is simple for some categories having clear user interactions, e.g. Kettle and Lamp, while is difficult for others having complex electrical consumption signatures, e.g. Computer Station and Laptop. In the uncertain cases, we decide to label only the User Interaction events on which we are confident.

We divide the appliances by using the categorization presented above. Hi-Fi system and Fan have problems of *interaction without transitions*, while Mobile phone has problems of *transition selection*. The signatures belonging to these three classes are treated with the rule-based system.
We compute the accuracy rate in three different tests:

1. When the correct model and the real sequence of states are known a priori. In this case we measure the performance of our system making the hypothesis that the HMM is performing a perfect Viterbi alignment with a perfect appliance class recognition. We refer to this system as Full Oracle test.

2. When only the correct model is known. In this case we measure the performance of our system making the hypothesis of a HMM performing a perfect appliance class recognition and we use the computed Viterbi alignment. We refer to this system as Category Oracle test.

3. With no a priori information. In this case we use the computed appliance class recognition and Viterbi alignment. We refer to this system as Regular test.

For each test we compute the ability of the system to detect the User Interaction events. We consider the detection correct when the elapsed time between the real and the detected interaction is under 30 seconds (corresponding to 3 samples distance with the sampling frequency of 10^{-1} Hz). For the misclassifications we separate false positive (FP) and false negative (FN). In the case of a FP, an interaction is not detected when it occurs, in the case of FN, an interaction is detected when not present. We perform the test following the *Intersession* protocol, the simplest between the two described with the ACS-F database.

3.9.4 Results and Discussion

As first step, we use HMMs to determine the appliance class identifications and state sequences. This information is sent to the User Interaction Layer that recovers the User Interaction events. As a next step, we send the test data and HMMs outputs to the User Interaction Layer to perform the three tests presented in previous section. In Table 3.20 we show the results obtained after the User Interaction Layer.

In the Full Oracle test we obtain an accuracy rate of 96.3%. Having the a priori information about the correct model and the sequence of states, the *User Interaction Layer* performs rather well. Several classes attain the 100% of accuracy rate. In the Category Oracle test we obtain an accuracy rate of 82.5%. We notice a drop in the performances when we let the HMM compute the alignment with the Viterbi algorithm. A small error rate in the state recognition has a great impact in the accuracy rate of the User Identification events. This is explained by the importance of recovering the correct

		Full Oracle			Category Oracle			Regular		
class	**type**	TP	FP	FN	TP	FP	FN	TP	FP	FN
Hi-fi	UDA	49	3	3	37	15	17	37	15	17
Television	UDA	40	3	0	38	5	3	38	5	3
Computer	UDA	22	5	2	22	5	0	22	5	0
Lamp Inc.	UDA	54	0	0	54	0	0	54	0	0
Lamp CFL	UDA	58	0	0	57	1	1	57	1	1
Fan	UDA	107	4	0	106	5	0	106	5	0
Monitor	UDA	62	0	0	61	0	5	61	1	4
Shaver	UDA	155	0	0	152	3	13	149	6	13
Coffee M.	FOA	39	0	0	34	5	7	34	5	7
Oven	FOA	59	0	0	59	0	4	59	0	4
Printer	FOA	45	0	0	37	8	26	37	8	28
Kettle	FOA	55	0	0	55	0	0	55	0	0
Mobile P.	BBA/UDA	16	4	5	10	10	3	10	10	3
Laptop	BBA/UDA	23	2	0	13	12	4	13	12	17
Fridge	TCA/UDA	14	0	0	9	5	0	9	5	0
Total		798	21	10	744	74	83	741	78	97

Table 3.20: Results in terms of number of events obtained when applying the *User Interaction Layer* for the three tests. Full Oracle test when the correct model and the real sequence of states are known a priori, Category Oracle test when only the correct model is known and Regular test with no a priori information. UDA stands for Usage dependent appliances, FOA for fixed operation appliances, TCA for Thermostatically controlled appliances and BBA for Battery based appliances.

state sequence for the computation of the state transitions. In fact, it directly influences the User Interaction events detection through the state transitions phase. In the Regular test we obtain an accuracy rate of 80.5%. We notice a small deterioration in performances compared to the previous case. The difference is small because the accuracy rate computed on the appliance detection is quite high and misclassifications among similar models provide similar sequences of states.

Finally we notice that our system of rules improves the results for the class on which it has been applied: in the three tests we have a mean improvement of 25.8% for the Hi-Fi class, of 16.5% for the Mobile phone class and of 18% for the Fan class. In particular the rules related to the *Interaction without transition* improve the FP, while those related to the *Transition selection* improve the FN.

We remark that some classes are more challenging than others, as Hi-Fi, Computer and Laptop. This observation can be explained by the fact that the appliances belonging to these classes require high level of interaction with the User and during the labeling phase the errors are more frequent. Finally we observe that the Mobile phone, even after the application of the rules, remains a difficult class for the event identification. Generally speaking, the UDA devices seem to better degrading when compared to the other categories. With the exception of the Hi-fi and Shaver, the step from the Full Oracle to the Category Oracle tests does not appear dramatic. Instead, the FOA devices apparently have more difficulties when performing the Category Oracle and Regular tests, with the exception of the Kettle category which always yield a perfect User Interaction recognition. The other three categories, i.e. Mobile phone, Laptop and Fridge, which belong to two different categories (BBA/TCA and UDA), present several problems when performing the Category Oracle and Regular tests.

3.10 Conclusion

In this chapter, we reported about our works in the field of Intrusive Load Monitoring (ILM) for the appliance recognition. Given that this field was not intensively explored in literature, we elaborated a new proposition for the ILM categorization. We proposed to subdivide ILM in three categories: ILM I when zones of the house are monitored through energy meters placed at the circuit breaker level, ILM II when a group of appliances is monitored through energy meters placed at the plug level and

ILM III when the appliance energy consumption is monitored through energy meters placed at the appliance level.

We noticed that researchers have the tendency to record and use their own data, leading to a situation where only few data-sets were publicly available. In order to answer to this need, we created our own database, called ACS-F, and we made it freely available to the scientific community. Two versions have been created: the first version, called ACS-F1, contains 100 appliances uniformly spread among 10 categories, while the second version, called ACS-F2, contains 225 different appliances uniformly spread among 15 categories. The appliances were recorded at low frequency in two sessions of one hour, called *session 1* and *session 2*. All appliances contained in the database are from different brands and/or models, making possible the computation of generic models.

By comparing in details ILM papers, we observed numerous differences between the analysis approaches, making difficult any comparison of the researchers' results. We observed important differences between the type of appliances, the feature extraction and the machine learning techniques used. In order to make possible a fair comparison between the appliance identification techniques, we created several analysis protocols with different levels of difficulty. We proposed two basic protocols, called *intersession* and *unseen appliance* protocol. The first one classifies appliances already seen in the training set, while the opposite occurs for the second protocol. We proposed also two protocols for comparing the performances of the appliance identification when using portions of signal shorter than one hour.

We analyzed and evaluated different strategies of machine learning fitting to these protocols. We performed a pre-processing phase, where we added to the original feature space the information about their evolution through the computation of dynamic coefficients. Data was normalized using the z-score normalization. We proposed three machine learning techniques for the appliance identification: K-NN, GMMs and HMMs. K-NN is a non-parametric discriminative approach, while GMMs and HMMs have a parametric generative approach. By comparing the accuracy rates, we observed that GMMs and HMMs outperform K-NN, while GMMs and HMMs provide similar results. Only when using the *unseen appliance* protocol on the ACS-F2, we observed better performances of HMMs compared to GMMs. In order to relax the observation independence hypothesis standing behind GMMs/HMMs, we added the dynamic coefficients. We systematically compared the accuracy rates when adding the dynamic coefficients and we found that their inclusion is beneficial for GMMs and HMMs, in particular when using the *unseen appliance* protocol. In order to detect the most important features we also performed a feature selection leading significant improvements for the K-NN. More limited improvements were observed for GMMs and HMMs, that seem to be less sensitive to non relevant features.

We analyzed the scalability property of generative models. In a first application we computed the evolution of the accuracy rate of a GMM when reducing the number of classes. This computation was performed on all possible appliance combinations, by simply comparing the scores computed through the generative algorithm. A second interesting application was the multi-signal identification. In this case we tried to identify appliances using signatures containing more than one appliance at a time. Instead of training specific models on group of appliances, we proposed an alternative method consisting in merging the Gaussians generated by the single models. By using this method, we obtained promising performances in terms of computational time and accuracy rate. A third application was based on the distribution of the computational resources using the "natural distributivity" of the computation allowed by generative models. We introduced the concept of *Virtual Sensor* and *Virtual Class*. The first is non-physical sensor that appears as a regular one within the network, while the second contains the trained models. The *Virtual Class* is able to compute a likelihood for each signature and the *Virtual Sensor* collect and compare the likelihoods coming from the *Virtual Classes*. When inferring the priors, it is therefore able to select the winner.

Finally, we showed an additional advantage in using state-base modeling for the appliance recognition. In the appliance signature modeling, the state transitions usually correspond to a human interaction. Therefore we tried to retrieve the User Interactions from the Viterbi alignment. The system showed some limitations, i.e. when User Interactions are verified inside a given state or when the state transitions are not related to User Interactions. We applied a rule-based system for improving our User Interaction recognition. We observed that small errors in the state alignment has a great impact on the accuracy rate of the User Identification events.

Bibliography

[1] "Wi-fi smart plug." `http://us.dlink.com/products/connected-home/wi-fi-smart-plug/`, 2015.

[2] "Wemo insight switch." `http://www.belkin.com/us/F7C029-Belkin/p/P-F7C029/`, 2015.

[3] "The modlet." `http://www.thinkecoinc.com/products/the-modlet/`, 2015.

[4] M. Weiss, T. Staake, D. Guinard, and W. Roediger, "eMeter: An interactive energy monitor," *Higher Education*, pp. 3–4, 2009.

[5] D. Guinard, M. Weiss, and V. Trifa, "Are you Energy-Efficient: Sense it on the Web!," tech. rep., Institute for Pervasive Computing, ETH Zürich, Zürich, 2009.

[6] C. Gisler, G. Barchi, G. Bovet, A. Mugellini, and J. Hennebert, "Demonstration Of A Monitoring Lamp To Visualize The Energy Consumption In Houses," in *Proceedings of the 10th International Conference on Pervasive Computing*, 2012.

[7] S. Darby, "The Effectiveness Of Feedback On Energy Consumption - A review for DEFRA of the literature on metering, billing, and direct displays," *Environmental Change Institute, University Of Oxford*, vol. Technical, no. April, p. 21, 2006.

[8] A. Ridi, C. Gisler, and J. Hennebert, "A survey on intrusive load monitoring for appliance recognition," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[9] F. Sultanem, "Using Appliance Signatures for Monitoring Residential Loads ot Meter Panel Level," *IEEE Transactions on Power Delivery*, vol. 6, no. 4, pp. 1380–1385, 1991.

[10] G. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[11] A. Marchiori, D. Hakkarinen, Q. Han, and L. Earle, "Circuit-Level Load Monitoring for Household Energy Management," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 40–48, 2011.

[12] L. Harrington, J. Brown, S. Holt, A. Meier, B. Nordman, and M. Ellis, "Standby Energy : building a coherent international policy framework – moving to the next level," *Summer Study Proceedings, European Council for an Energy Efficient Economy*, vol. 3, pp. 1285–1296, 2007.

[13] J. Ross and L. Berkeley, "Whole-House Measurements of Standby Power Consumption," in *Proceedings of the 2nd International Conference on Energy Efficiency in Household Appliances (EEDAL)*, 2000.

[14] T. Cease and P. Johnston, "A magneto-optic current transducer," *Power Delivery, IEEE Transactions on*, vol. 5, pp. 548–555, Apr 1990.

[15] A. Ridi, C. Gisler, and J. Hennebert, "User interaction event detection in the context of appliance monitoring," in *Proceedings on the IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM Workshops)*, pp. 323–328, March 2015.

[16] A. Zoha, A. Gluhak, M. Imran, and S. Rajasegarar, "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[17] M. Figueiredo, A. De Almeida, and B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," *Neurocomputing*, vol. 96, pp. 66–73, 2012.

[18] M. B. Figueiredo, A. De Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (NILM) systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6594 LNCS, pp. 31–40, 2011.

[19] W. Chan, a.T.P. So, and L. Lai, "Harmonics load signature recognition by wavelets transforms," in *Proceeding of the International Conference on Electric Utility Deregulation and Restructuring and Power Technologies.*, pp. 4–7, 2000.

[20] S. B. Leeb, J. L. Kirtley, M. S. Levan, and J. P. Sweeney, "Development and Validation of a Transient Event Detector," *AMP Journal of Technology*, vol. 3, pp. 69–74, 1993.

[21] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of the 26th Conference on Artificial Intelligence*, pp. 356–362, July 2012.

[22] P. Armstrong, C. Laughman, S. Leeb, and L. Norford, "Detection of Rooftop Cooling Unit Faults Based on Electrical Measurements," *HVAC&R Research*, vol. 12, no. 1, pp. 151–175, 2006.

[23] R. Cox, *Minimally Intrusive Strategies for Fault Detection and Energy Monitoring.* PhD thesis, MIT, September 2006.

[24] Y.-C. Chen, C.-M. Chu, S.-L. Tsao, and T.-C. Tsai, "Detecting users' behaviors based on nonintrusive load monitoring technologies," in *Proceedings of the 10th IEEE International Conference on Networking, Sensing and Control*, pp. 804–809, 2013.

[25] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging smart meter data to recognize home appliances," in *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 190–197, 2012.

[26] L. Jiang, S. Luo, and J. Li, "An approach of household power appliance monitoring based on machine learning," in *Proceedings of the 5th IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pp. 577–580, 2012.

[27] L. Robert and K. Liszewski, "Methods of electrical appliances identification in systems monitoring electrical energy consumption," in *Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, pp. 10–14, 2013.

[28] M. Berges, E. Goldman, H. Matthews, and L. Soibelman, "Training load monitoring algorithms on highly sub-metered home electricity consumption data," *Tsinghua Science & Technology*, vol. 13, pp. 406–411, Oct. 2008.

[29] F. Paradiso, F. Paganelli, A. Luchetta, D. Giuli, and P. Castrogiovanni, "ANN-based appliance recognition from low-frequency energy monitoring data," in *Proceedings of the 14th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–6, June 2013.

[30] Y. Huang and Y. Lai, "Distributed energy management system within residential sensor-based heterogeneous network structure," *Wireless Sensor Networks and Ecological Monitoring*, vol. 3, pp. 35–60, 2013.

[31] A. Reinhardt, D. Burkhardt, M. Zaheer, and R. Steinmetz, "Electric appliance classification based on distributed high resolution current sensing," in *Proceedings of the 7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, pp. 1003–1009, 2012.

[32] "Plugwise." `http://www.plugwise.com/`, 2010.

[33] K. Basu, V. Debusschere, and S. Bacha, "Appliance usage prediction using a time series based classification approach," in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON)*, pp. 1217–1222, 2012.

[34] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, "Design and implementation of a high-fidelity ac metering network," in *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 253–264, 2009.

[35] H. Serra, J. Correia, A. Gano, A. de Campos, and I. Teixeira, "Domestic power consumption measurement and automatic home appliance detection," in *Proceedings of the 4th IEEE International Workshop on Intelligent Signal Processing (WISP)*, pp. 128–132, 2005.

[36] M. Marcu and A. Stancovici, "Systems classification based on power signatures," in *Proceedings of the 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT)*, pp. 1–6, Dec. 2011.

[37] A. Schoofs, A. Ruzzelli, and G. O'Hare, "Appliance activity monitoring using wireless sensors," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 434–435, 2010.

[38] A. Schoofs, A. Guerrieri, D. Delaney, G. O'Hare, and A. Ruzzelli, "ANNOT : Automated Electricity Data Annotation Using Wireless Sensor Networks," in *Proceedings of the 7th Annual IEEE Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 1–9, 2010.

[39] R. Prasad and S. Semwal, "A Simplified New Procedure for Identification of Appliances in Smart Meter Applications," in *Proceedings of the 7th IEEE International Systems Conference (SysCon)*, pp. 339–344, 2013.

[40] S. Lee, G. Lin, W. Jih, and J. Hsu, "Appliance Recognition and Unattended Appliance Detection for Energy Conservation," in *Proceeding of the 24th Conference on Artificial Intelligence (AAAI)*, pp. 37–44, 2010.

[41] S. Lee, G. Ryu, Y. Chon, R. Ha, and H. Cha, "Automatic Standby Power Management Using Usage Profiling and Prediction," *IEEE Transactions on Human-Machine Systems*, vol. 43, pp. 535–546, Nov. 2013.

[42] H. Cho and M. Hahn, "Simple and Robust Method for Detecting the Electrical Appliances Using Markers and Programmable Logic Devices," in *Proceedings of the 13th IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 334–338, 2009.

[43] H. Cho, T. Yamazaki, and M. Hahn, "Determining Location of Appliances from Multi-hop Tree Structures of Power Strip Type Smart Meters," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2314–2322, 2009.

[44] T. Kato, H. Cho, D. Lee, T. Toyomura, and T. Yamazaki, "Appliance recognition from electric current signals for information-energy integrated network in home environments," *Ambient Assistive Health and Wellness Management in the Heart of the City*, vol. 5597, pp. 150–157, 2009.

[45] C. Gisler, A. Ridi, D. Zufferey, O. A. Khaled, and J. Hennebert, "Appliance consumption signature database and recognition test protocols," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications (Wosspa '13)*, pp. 336–341, 2013.

[46] A. Ridi, C. Gisler, and J. Hennebert, "Unseen appliances identification," in *Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP)*, pp. 75–82, 2013.

[47] H. Najmeddine, K. El Khamlichi Drissi, C. Pasquier, C. Faure, K. Kerroum, A. Diop, T. Jouannet, and M. Michou, "State of art on load monitoring methods," in *Proceedings of the 2nd IEEE International Power and Energy Conference*, pp. 1256–1258, 2008.

[48] L. Duarte, E. Ferreira, and J. Dias, "Energy Measurement Techniques for Energy Efficiency Programs," in *Energy Efficiency – The Innovative Ways for Smart Energy, the Future Towards Modern Utilities* (M. Eissa, ed.), InTech, Chapters published, 2012.

[49] M. Ito, R. Uda, S. Ichimura, K. Tago, T. Hoshi, Y. Matsushita, and K. Hachioji, "A Method of Appliance Detection Based on Features of Power Waveform," in *Proceedings of the 4th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, pp. 291–294, 2004.

[50] G. Bovet, A. Ridi, and J. Hennebert, *Toward Web Enhanced Building Automation Systems - Big Data and Internet of Things: A Roadmap for Smart Environments*, vol. 546 of *Studies in Computational Intelligence*, ch. 11, pp. 259–284. Springer, 2014.

[51] M. Baranski and J. Voss, "Non-intrusive appliance load monitoring based on an optical sensor," in *Proceedings of the IEEE Bologna Power Tech Conference*, vol. 4, pp. 267–274, 2003.

[52] F. Zaidi Adeel Abbas and P. Palensky, "Load Recognition for Automated Demand Response in Microgrids," in *Proceedings of the 36th IEEE Annual Conference on Industrial Electronics Society (IECON)*, pp. 2442–2447, 2010.

[53] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the Accuracy of Appliance Identification Based on Distributed Load Metering Data," in *Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability*, 2012.

[54] H. Lam, G. Fung, and W. Lee, "A Novel Method to Construct Taxonomy of Electrical Appliances Based on Load Signatures," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, 2007.

[55] D. Zufferey, C. Gisler, O. Abou Khaled, and J. Hennebert, "Machine learning approaches for electric appliance classification," in *Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pp. 740–745, 2012.

[56] A. Ridi, C. Gisler, and J. Hennebert, "Automatic identification of electrical appliances using smart plugs," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)*, pp. 301–305, 2013.

[57] T. Saitoh, Y. Aota, T. Osaki, R. Konishi, and K. Sugahara, "Current Sensor based Non-intrusive Appliance Recognition for Intelligent Outlet," in *Proceedings of the 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 349–352, 2008.

[58] F. Englert, T. Schmitt, S. Kößler, A. Reinhardt, and R. Steinmetz, "How to Auto-Configure Your Smart Home? High-Resolution Power Measurements to the Rescue," in *Proceedings of the 4th International Conference on Future energy systems (E-ENERGY)*, pp. 215–224, 2013.

[59] S. Kang, K. Shin, R. Yang, S. Lee, and J. Sohn, "Cost effective disaggregation mechanism for the NIALM system," *International Journal of Smart Grid and Clean Energy*, vol. 2, no. 3, pp. 364–370, 2013.

[60] J. Kolter and M. Johnson, "REDD : A Public Data Set for Energy Disaggregation Research," in *Proceedings of the ACM Workshop on Data Mining Applications in Sustainability (SUSTKDD)*, 2011.

[61] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. Bajic, "AMPds : A Public Dataset for Load Disaggregation and Eco-Feedback Research," in *IEEE Electrical Power and Energy Conference (EPEC)*, 2013.

[62] A. Kyle, A. Filip, D. Benítez, D. Carlson, A. Rowe, and M. Bergés, "BLUED : A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research," in *Proceedings of the 2nd ACM Workshop on Data Mining Applications in Sustainability (SUSTKDD)*, 2012.

[63] A. Monacchi, D. Egarter, W. Elmenreich, S. D'Alessandro, and A. Tonello, "GREEND: An Energy Consumption Dataset of Households in Italy and Austria," in *Proceedings of IEEE International Conference on Smart Grid Communications*, pp. 1–16, May 2014.

[64] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, 2015.

[65] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart *: An Open Data Set and Tools for Enabling Research in Sustainable Homes," in *Proceedings of the 2nd ACM Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2012.

[66] N. Batra, M. Gulati, A. Singh, and M. Srivastava, "It's different: Insights into home energy consumption in india," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pp. 3:1–3:8, ACM, 2013.

[67] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. Paradiso, "A Platform for Ubiquitous Sensor Deployment in Occupational and Domestic Environments," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 119–127, 2007.

[68] E. Tapia, S. Intille, L. Lopez, and K. Larson, "The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection," in *Proceedings of the 4th International Conference on Pervasive Computing (PERCOM)*, 2006.

[69] J. Gao, S. Giri, and E. Kara, "Demo Abstract : PLAID : a Public Dataset of High-Resolution Electrical Appliance Measurements for Load Identification," in *1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings*, 2014.

[70] J. Zimmermann, "End-use metering campaign in 400 households In Sweden Assessment of the Potential Electricity Savings," tech. rep., Enertech, 2009.

[71] A. De Almeida and P. Fonseca, "Residential Monitoring to Decrease Energy Use and Carbon Emissions in Europe," in *Proceedings of the 4th International Energy Efficiency in Domestic Appliances and Lighting Conference (EEDAL)*, 2006.

[72] N. Piotrowski, "True Power vs . Apparent Power : Understanding the Difference." `http://www.aptsources.com/resources/pdf/True%20vs.%20Apparent%20Power.pdf`, 2010.

[73] J. Liang, S. Ng, G. Kendall, and J. Cheng, "Load Signature Study Part I: Basic Concept, Structure, and Methodology," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, 2010.

[74] S. Phelan, C. McArdle, S. Daniels, and P. Meehan, "Temporal and frequency analysis of power signatures for common household appliances," in *Procedings of the Symposium on ICT and Energy Efficiency and Workshop on Information Theory and Security (CIICT)*, pp. 22–27, 2012.

[75] A. Reinhardt, D. Christin, and S. Kanhere, "Predicting the Power Consumption of Electric Appliances through Time Series Pattern Matching," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BUILDSYS)*, 2013.

[76] M. Fitta, S. Biza, M. Lehtonen, T. Nieminen, and G. Jacucci, "Exploring Techniques for Monitoring Electric Power Consumption in Households," in *Proceedings of the 4th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, pp. 471–477, 2010.

[77] W. Lee, G. Fung, H. Lam, F. Chan, and M. Lucente, "Exploration on Load Signatures," in *Proceedings of the International Conference on Electrical Engineering (ICEE)*, 2004.

[78] A. Ridi, C. Gisler, and J. Hennebert, "ACS-F2 - A New Database of Appliance Consumption Analysis," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SOCPAR)*, 2014.

[79] H. Gonçalves, A. Ocneanu, and M. Bergés, "Unsupervised disaggregation of appliances using aggregated consumption data," in *Workshop on Data Mining Applications in Sustainability (SUSTKDD)*, 2011.

[80] J. Hennebert, *Hidden Markov models and artificial neural networks for speech and speaker recognition.* PhD thesis, EPFL, Lausanne, October 1998.

[81] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *Proceedings of the 2009 ASCE International Workshop on Computing in Civil Engineering*, (Austin, USA), 2009.

[82] A. Biem, "A Model Selection Criterion for Classification: Application to HMM Topology Optimization," in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, 2003.

[83] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of 20th International Conference on Machine Learning (ICML)*, 2003.

[84] "Sum of normally distributed variables." `https://en.wikipedia.org/wiki/Sum_of_normally_distributed_random_variables`, 2015.

[85] A. Ridi, C. Gisler, and J. Hennebert, "Appliance and State Recognition using Hidden Markov Models," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, 2014.

[86] G. Bovet, A. Ridi, and J. Hennebert, "Virtual things for machine learning applications," in *Proceedings of the 5th International Workshop on Web of Things*, WoT, pp. 4–9, ACM, 2014.

[87] G. Bovet, A. Ridi, and J. Hennebert, "Appliance recognition on internet-of-things devices," in *Proceedings of the 4th International Conference on Internet of Things, Demonstration Session*, IoT, 2014.

Chapter 4

Glaucoma detection

Contents

4.1 Introduction

As a matter of facts, our society is aging, resulting in increasing costs for hospitals, medical practitioners and health insurers. In addition, many people with disabilities or suffering chronic illnesses require repeated hospital visits and admission due to recurrent symptoms. Unavoidably, during the hospitalization the quality of life of the patient is reduced. Health Smart Homes (HSH) have been proposed as solution for such problem, providing health care services for people living in their own home [1]. The principal health care service provided by HSH is the continuous monitoring of the health status, defined as the *complete physical, mental and social well-being and not merely the absence of disease or infirmity over a period of time* [2]. For checking the health of a patient, information about his physical and mental status has to be collected. This task can be achieved by capturing and analyzing physiological data.

How in practice physiological data is obtained? There are mainly 3 approaches, namely (i) direct paper based on web questionnaires, (ii) vision based approaches and (iii) wearable sensors. The first approach has several advantages, as the fast data gathering, the minimal costs and the ease of data automation. However, there are two important drawbacks: this approach is limited to "informative"

and not-measured data, and the truthfulness of the information depends on honesty of the person who fill the web-questionnaire. Vision-based approaches analyze video stream coming from regular, stereo or infrared cameras. In the health monitoring field, they are often used for detecting fall events or retrieve information as body temperature, posture, etc. The major drawbacks are the computational load, the sensitivity to illumination variations and the privacy concerns. Wearable sensors-based approaches use devices directly placed on human body (surface sensors), in human body (implanted sensors) or near human body, e.g. carried in clothes pockets, by hand or in bags. Physiological data captured by wearable sensors can be directly sent to the medical practitioners. The major drawback is that patients have to wear, install, care and ware the sensors that are sometimes heavy or cumbersome. The set of devices and technology used for monitoring specific parts of the human body through wearable sensors is also called wearable health-monitoring system (WHMS).

In the last decades, patients are more and more monitored at home using WHMS. Usually WHMS are addressed to people that need to be constantly monitored or that require regular drug administration, as elderly people, people suffering chronic illnesses or with disabilities. WHMS are becoming more and more popular because of the reduction of the size and weight of sensors and also thanks to the democratization of wireless and Internet connectivity. In recent trends, sensors are becoming more and more pervasive such as in *textile sensors*, also called *smart garments*, that are integrated into textile fibers [3].

In our work, we will focus on machine learning techniques based on stochastic modeling applied to physiological signals acquired with WHMS. In particular, we will define the physiological signal in Section 4.2.1 and we will review several works in literature that use physiological signals in HSH (Section 4.2.2), focusing on in-home health monitoring (Section 4.2.3). Given the data availability, we pragmatically focus on a specific domain, i.e. the recognition of the glaucoma illness through the monitoring of eye volume variations (Section 4.3). In particular, we use data from a wearable contact lens sensor, called Triggerfish®, developed by Sensimed SA. We will present our works concerning the feature extraction (Section 4.4) and the application of generative machine learning techniques for the recognition of the glaucoma illness (Section 4.5). Finally in Section 4.6 we will conclude the chapter.

4.2 State of the art

4.2.1 Physiological signals

Physiological signals have been defined as [4]:

Observations of physiological activities of organisms, ranging from gene and protein sequences, to neural and cardiac rhythms, to tissue and organ images.

Even if not strictly equivalent, the term "biomedical signal" is often used in the literature as a synonym of "physiological signal". Physiological signals have three main characteristics: (i) **accessibility**, (ii) **physical quantity** and (iii) **support signal** [5]. The **accessibility** is the "ability to access" to the signal. Three categories exist: *internal* when a device is placed "on" the body in order to capture an internal signal, *emanated* when the signal emitted from the body is captured with external devices, e.g. cameras capturing infrared radiations, and *derived* when analysis are performed on samples of tissue derived from the human body, e.g. blood test. The measured **physical quantity** depends on the biological system under analysis and it belongs to one among the following categories: biopotential, pressure, flow, dimensions, displacement, impedance, temperature, chemical concentration and composition. In order to be easily processed, the physical quantity is usually transformed in an electrical signal. This operation is performed by an active or passive transducer. Active transducer, e.g. piezoelectric, converts a physical quantity in electric current without requiring any power source. Passive transducer, e.g. strain gauge, produces variations in terms of an electrical quantity, such as capacitance, resistance, or inductance. Finally, depending on the **support signal**, the physiological signals can be divided in [6]:

- *Bio-electrical* signals are generated by cells where an electrical potential is conducted, typically the nerve and muscle cells. The variation of the electrical potential of these cells generates a

current that can be measured. Electrocardiography (ECG), Electromyography (EMG) and Electroencephalography (EEG) are examples of different processes recording an electrical activity.

- *Bio-mechanical* signals are produced by mechanical movements of the human body. In terms of physical quantity, bio-mechanical signals are related to the motion, displacement, pressure and flow of the human body or its sub-systems. For instance the Respiratory Volume (RV) is related to the movement of the chest and for this reason it is considered a bio-mechanical signal.

- *Bio-acoustic* signals are typically produced by flows in the human body. Special microphones are able to capture the sound produced by these flows, e.g. the Phonocardiogram (PCG) captures the sound waves generated by the blood flow in the heart.

- *Bio-magnetic* signals are produced by the magnetic field generated together with the electrical field.

- *Bio-chemical* signals are measured from tissues of the human body, as the concentration of a blood constituent.

- *Bio-impedance* signals measure the impedance of a certain tissue to the electrical current. Typically the electrodermal activity (EDA) measures the conductance of the skin. EDA is also known with other names, as skin conductance (SC), galvanic skin response (GSR) or electrodermal response (EDR) depending on the field of application.

- *Bio-optical* signals are the answer of optical functions to biological systems. For example the Photoplethysmography (PPG) measures the transmitted and reflected light from the blood vessel.

As we have seen, the definition of physiological signal is very large and several different types of signals are included. Consequently, various health-monitoring systems can be applied to the human body, as shown in Figure 4.1. In this chapter we focus on wearable devices directly placed on the body, thus only internal signals have been considered. External devices, as cameras, are not considered in this work.

4.2.2 Physiological signal applications

Physiological signals are used in different contexts:

- **Health monitoring**. A review of the papers dealing with wearable sensor for health monitoring is provided by Pantelopoulos et al. [7]. Health monitoring is usually addressed to elderly people, people suffering chronic illnesses or people with disabilities. We may divide the health monitoring applications depending on the human sub-system under observation: general health, cardiovascular, brain, respiratory and visual monitoring. More information will be presented in Section 4.2.3.

- **Activity recognition**. A review of activity recognition applications is presented by Lara et al. [8]. This paper shows the large variability of approaches based on wearable sensors. Some physiological signals, as the heart rate, have a little discriminatory power because of the physical strain consequences. In fact, after high-demanding activities, the heart rate signal remains at high level for a long time, even if the following activity is low-demanding [9]. In particular, physiological signals coming from the heart rate and respiration correlate with the intensity of the activity and not the type of the activity, therefore it can be associated with difficulty to a precise activity [10]. Some researchers have developed special feature extraction techniques able to deal with the vital sign stabilization in order to avoid the problem of the stabilization of the heart rate signal [11]. The activity recognition appears easier when using physiological signals that are not strictly related to the physical effort. For instance, the electrooculographic signal is currently used to recognize office-related activities, as reading, browsing, writing, watching video, etc. In this direction, Bulling et al. [12] try to detect five different activities by performing a feature extraction able to retrieve several characteristics related to three physiological aspects: saccades, fixation and blinks.

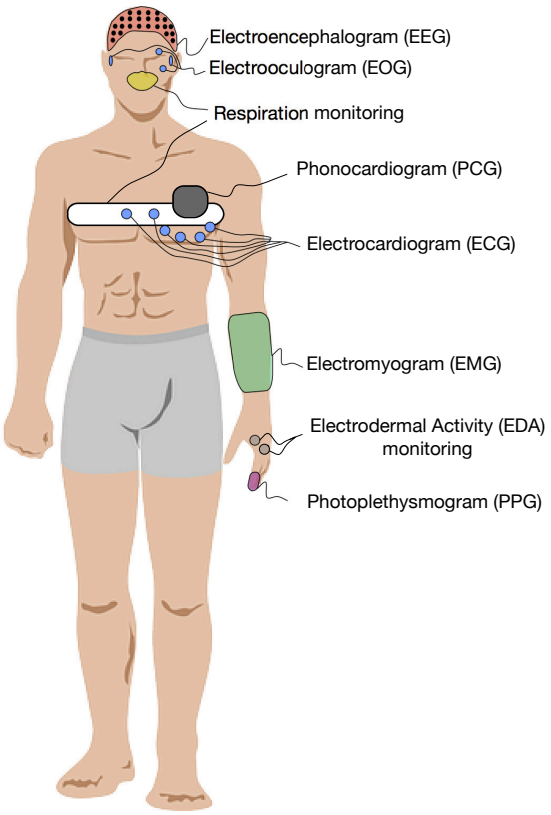


Figure 4.1: Several wearable health-monitoring systems applied to the human body are here represented, i.e. Electroencephalogram, Electrooculogram, Respiration monitoring, Phonocardiogram, Electrocardiogram, Electromyogram, Electrodermal activity monitoring and Photoplethysmography. The list of health-monitoring systems here represented is not exhaustive.

- **Emotion recognition**. A review of papers concerning this field has been presented by Jeritta et al. [13]. Physiological signals have been widely used to recognize emotions [14, 15]. While a person can hide emotions by controlling the external appearance, e.g. the facial expression, physiological patterns inevitably change and can be detected through the appropriate instruments [16]. Several emotions can be analyzed depending on the level of arousal and the pleasantness. Emotions influence several human sub-systems, from which physiological signals can be used for the emotion recognition, such as: *cardiovascular system*, *electrodermal activity*, *respiratory activity*, *muscular system* and *brain activity* [13]. When experiencing emotions, the sympathetic nervous system raises the hearth rate and the blood pressure and decreases the heart rate variability. Variations in the blood pressure cause variations in the blood flow and skin temperature [17, 18]. At the same time, the autonomic nervous system, that directly controls sweat glands in the skin, typically rises the level of perspiration [14, 19]. The respiration changes in response to emotions, typically increasing its frequency, in particular if the emotions are fear or anxiety [20]. Muscular activities imperceptible to the human eye are often produced, as facial expressions [21]. Finally, specific areas of the brain are activated depending on the positive or negative feelings [22] and different subjects experiencing the same emotions activate similar areas of the brain.

- **Biometric application**. Several physiological signals acquired from a person are specific to the person who generated them. This means that the person leaves a sort of "signature" on the generated signals. Biometric applications can therefore *identify* a person among several others or can *authenticate* a specific subject. Physiological signals can therefore be used for

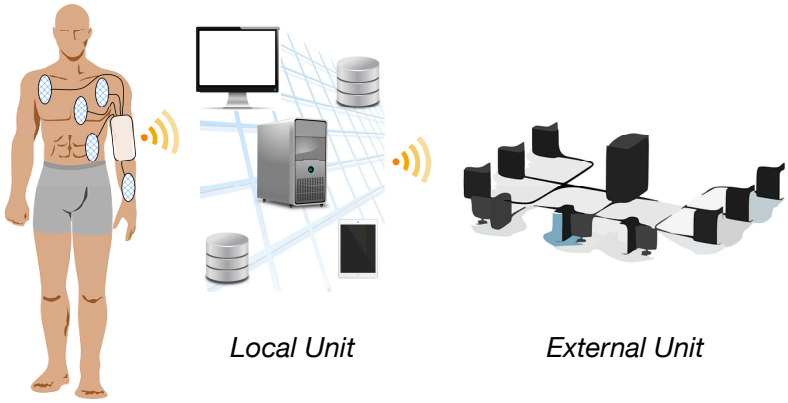


Figure 4.2: General representation of a wearable health-monitoring systems. It is composed of four parts: the *human body*, the *sensors*, the *local unit* and the *external unit*. The local unit is localized close-by the patient for data storage and basic signal processing. The external unit is potentially remote, typically in a medical center where the data is received and further processed.

biometric applications. Electrocardiographic (ECG) signals have been proven to be useful for biometric tasks [23, 24]. In particular ECG has been proposed for securing the communication in a WHMS, where several patients are simultaneously under observation and patient identity must be often verified [25]. Photoplethysmographic (PPG) signal has also been proven to be useful for biometric tasks [26, 27]. Electromyographic (EMG) signals on biometric application have also been studied in this field [28, 29].

4.2.3 In-home health monitoring

As illustrated in Figure 4.2, a WHMS is typically composed of four parts: the *human body*, the *sensors*, the *local unit* and the *external unit*. The human body represents the source of the physiological signals that are acquired through the sensors directly placed on the body. The local unit is placed in the proximity of the user and it manages the physiological signals acquired through the sensors. A wireless technology, e.g. Bluetooth or ZigBee, typically connects the sensors with the local unit. Several operations can be managed by the local unit, as data storing, data visualization, analysis of data and decision making, e.g. drug administration or sending an alarm. The external unit is typically located in a medical center. Health workers are usually able to receive in real time or with a short delay data recorded by the local unit. Usually a communication network as GSM, UMTS or Internet, connects the local unit with the external unit. In certain circumstances the WHMS could differ, for instance data can be directly sent to health workers or it can be just locally stored.

Several works in the literature deal with health monitoring using physiological signals. From a survey of these works, we can divide WHMS applications depending on the goal of the monitoring:

- *General health monitoring*

  Different brackets of the population are interested in systems which monitor their general health conditions. These individuals do not suffer of a specific disease, but require a particular attention on their health condition, as elderly people, persons with disabilities, infants, etc. Elderly people and individuals with disabilities in some cases need a constant health monitoring. Lin et al. [30] create a system for monitoring elderly people by collecting physiological data as temperature, blood pressure and heart rate. The system consists in a mobile physiological device, a base station, a device for exchanging data and the network managing center. They test the system in nursing centers with 17 volunteers, showing positive feedbacks for such solutions. Monitoring systems are also popular for controlling the health status of infants. Differently from the others persons, infants do not explicitly provide feedback about discomfort or their health status. In some unfortunate cases, sudden unexplained death of children of less than one year of age happens without exact causes. This is called sudden infant death syndrome (SIDS) and it usually

occurs during sleep. Vests for infant with integrated sensors for monitoring physiological signals, as respiration and heart rate, have been created [31]. However, a systematic review of literature recently have pointed out that home monitoring has not been shown to be clearly effective in preventing SIDS and further research is ongoing [32].

- *Cardiovascular monitoring*

  In the European Union about 45% of deaths are related to cardiovascular diseases and more than the 20% has a chronic cardiovascular disease [33]. Elderly people are usually considered those that need to be monitored the most. In literature, several works on wearable and wireless ECG monitoring systems for elderly people can be found [34]. Cardiovascular disorders can be detected by using different physiological signals, however the ECG provides higher quantity and quality of information. Several different types of disorders are currently identified with ECG, however arrhythmia seems to be the most popular [35, 36]. In order to correctly interpret the ECG variability, accelerometers information are sometimes added as features [37]. Smart garments appear to be very popular for the detection of the ECG signal, that otherwise would require more cumbersome solutions. Several works integrate textile sensors in shirts, as [37, 38]. PPG and blood pressure sensors are also able to give complementary information to the ECG signal [39].

- *Brain monitoring*

  Many people worldwide have medical conditions affecting their cognitive ability. In most cases, cognitive deficits are due to intellectual disabilities, acquired brain injuries or neurodegenerative diseases as dementia. Among the adults the rate of persons with an intellectual disability ranges between 3-6 per thousand, while among children it ranges between 3-14 per thousand [40]. In the U.S., more than 5 million of persons live with a traumatic brain injury disability [41]. Tura et al. [42] develop a wearable device for brain-injured children in order to reduce the hospitalization time and improve the quality of life. The device is able to capture the respiration rate, the oxygenation level and the movements of the subject. In another work [43], they present a portable device for children with a learning disability. The device is able to monitor blood oxygen saturation, heart and respiration rates and patient activity. Home monitoring has been also applied for mental disorders as the bipolar disorder. For instance, Valenzy et at. [44] develop a wearable system designed for persons suffering of bipolar disorder. They use ECG, respirogram and body posture information to recognize four states: depression, mixed state, hypomania, and euthymia. Mahoney et al. [45] analyze the monitoring of persons with dementia at home from the ethical point of view. They provide some guidelines for searching to guarantee a helpful level of surveillance without infringing the personal dignity. Vidal et al. [46] propose to use eye tracking technology for the health monitoring. In fact, several neurological disorders and diseases are related to eye movements, as AIDS Dementia Complex, autism, schizophrenia and Alzheimer.

- *Respiratory monitoring*

  Respiratory monitoring systems at home are beneficial in particular for patients suffering of chronic respiratory diseases. Moreover, the condition of the patients in respiratory medicine is usually long term. Several non-invasive techniques have been developed in the context of in-home monitoring. However, predicting respiratory failures is quite a difficult task, considering also that they could happen in short term or long term with a random apparition.

  People affected by respiratory diseases could benefit of a respiratory monitoring system. In [47] patients suffering of asthma are equipped with a portable spirometer able to send the ventilatory values to a remote monitoring center. The goal is to detect early signs of asthmatic deterioration and send an alarm to the healthcare personnel. Rigau et al. [48] present a portable device based on forced oscillation technique, able to compute the respiratory resistance, reactance and reliability indices. Dinesen et al. [49] study the use of in-home monitoring among chronic obstructive pulmonary disease (COPD) patients, finding a significant reduction of the hospital admission rate. Silva Junior et al. [50] monitor COPD patients by measuring the maximum inspiratory pressure and the inspiratory time constant. Data is subsequently sent to a monitor center over Internet. Other works focus on monitoring of respiration activity while sleeping.

People that suffer of sleep-disorder breathing (SDB) and obstructive sleep apnea syndrome are at risk for hypertension, cardiovascular diseases and mortality. Nakayama-Ashida et al. [51] effectuate and epidemiological study of sleep for SDB in usual lifestyles using in-home monitoring.

- *Visual monitoring*

  Visual health monitoring using non-invasive methods requires intensive efforts and resources, therefore few researcher teams and companies are tackling this problem. There are two common non-invasive approaches: (i) smart glasses and (ii) smart lens. While smart glasses mainly uses cameras for visual health monitoring [52], smart lenses are directly in contact with the eye and they can more easily retrieve physiological signals. Different types of sensors can be mounted on contact lens for measuring physiological signals as eye movements, tear glucose concentration, corneal temperature, blood oxygen and intraocular pressure.

  Eye movements
  One of the first works in this field has been presented by Robinson [53], who obtained the position of the eye by using an alternating magnetic fields and a wired contact lens. More recently, a wireless contact lens has been developed by Kim et al. [54]. The position of the eye is obtained through a contact lens equipped with a magnet and circuit board equipped with two magnetoresistive sensors.

  Tear glucose concentration
  Several teams are developing contact lens able to detect the glucose level by using chemical reaction with tears. Badugu et al. [55] propose to use a contact lens using boronic acid-containing fluorophores (BAFs) for the glucose level monitoring. They study the variation of pH and polarity within the contact lens while changing the sugar concentration. They verify the suitability of this method for the continuous glucose monitoring. March et al. [56] use fluorescent contact lens equipped with glucose sensor and have started a clinical trial on five patients. They find that the glucose level is well tracked and the contact lens well tolerated, a promising result for future continuous health monitoring. Alexeev et al. [57] develop a glucose-sensing material, called "photonic crystal" and propose its integration in a contact lens or directly in the eye. The glucose concentration changes the color diffraction of the material across the entire visible spectral region, making simple to understand the glucose level. Recently other works have been presented [58, 59, 60] and in 2014 Google and Novartis started a collaboration on this subject [61].

  Corneal temperature
  Corneal temperature has been found related to tearing, inflammation, carotid artery disease and other factors [62]. Kinn et al. [63] develop a thermally sensitive contact lens that changes color depending on the temperature of the eye. The lens encapsulates liquid crystals that change the scatter of light depending on the temperature. By analyzing the color of the lens, the corneal surface temperature is retrieved.

  Blood oxygen
  Scott et al. [64] construct a contact lens probe able to measure pulse-oximetry on the blood in the retinal fundus. The lens includes a lenticular on the front surface focusing the light coming by an infrared and a red LED. A photo-diode captures the response of the retina to the light stimulation and generates the pulse-oximetry signal.

  Intraocular pressure
  The intraocular pressure (IOP) is the major risk factor for glaucoma, an eye conditions resulting in optic nerve damage and the second leading cause of blindness worldwide. IOP is a physiological signal rapidly changing in time and its evolution in time, called *continuous IOP monitoring*, is interesting for its relation with glaucoma. This point will be more developed in the next section.

4.2.4 Continuous monitoring of Intraocular pressure

More than 30% of the world population has vision defect as myopia. Some specific visual illnesses, i.e cataract, trachoma and glaucoma, are responsible of 70% of the global blindness worldwide [65]. Cataract is the loss of the normal transparency of the crystalline lens due to opacity and no prevention is known for this eye condition [66]. Trachoma is due to an infection, called Chlamydia trachomatis,

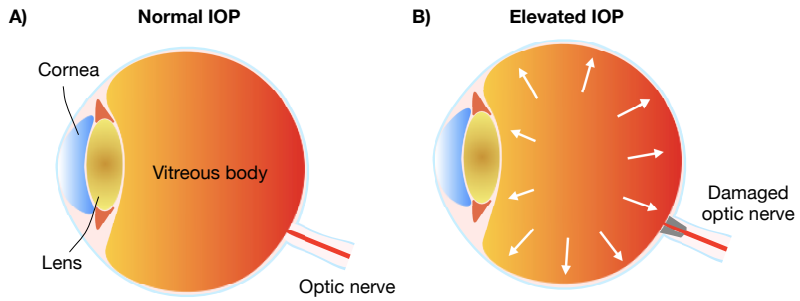


Figure 4.3: Comparison between a normal eyeball (a) and another with an elevated IOP (b). The elevated IOP, represented by the white arrows, causes damages on the optic nerve.

and it is usually related to inadequate hygienic living conditions. As previously said, glaucoma is the second leading cause of blindness worldwide and it is an eye conditions resulting in optic nerve damage. In 2010, more than 60 million people suffered from glaucoma and this number is expected to increase to 80 millions by 2020 [67]. Moreover, it has been shown that glaucoma is associated with a loss in the quality of life [68].

For a long time, IOP above 21 mmHg was considered as a direct indicator of glaucoma [69]. However, several patients developed glaucoma without showing an elevated IOP and others never developed glaucoma even with high IOP values. Nowadays, IOP is considered, together with age and the family history, as risk factors for the glaucoma [70], as illustrated in Figure 4.3. A glaucoma is declared when the optic nerve is damaged, often related to a high IOP.

Several types of glaucoma exist, yielding different symptoms and levels of severity:

- Primary Open-Angle Glaucoma (POAG) is the most common form of glaucoma associated with an open, normal appearing anterior chamber angle and raised IOP, with no other underlying diseases. "Open-angle" refers to the angle between the iris and the cornea that is as wide and open as it should be. This type is usually referred to just as "glaucoma" as it is the most common type.

- Angle-Closure Glaucoma (ACG) is a less common form of glaucoma. It has a closed or narrow angle between the iris and cornea, caused by blocked drainage canals.

- Normal-Tension Glaucoma (NTG) is a form of glaucoma where the optic nerve is damaged even if the IOP is not very high. However, patients with POAG and NTG show very similar eye conditions and the effective distinction between them is currently discussed by the scientific community [71, 72].

- Exfoliative Glaucoma (XFG) is caused by the abnormal accumulation of protein in the drainage system. It is more recurrent in Northern Europe.

- Many other types exist, as the Pigmentary Glaucoma, Traumatic Glaucoma, Congenital Glaucoma, Neovascular Glaucoma, Uveitic Glaucoma, etc.

Usually ophthalmologists take decisions about treatments only observing 1 or 2 measures of IOP during a clinical evaluation. An elevated IOP can be reduced by using treatments as drugs, laser and surgical therapy. Nowadays, it is well known that IOP varies over time and shows a 24-hours cycle, following the circadian rhythm. David et al. [73] analyze more than 2200 IOP curves recorded during the daytime and they observe that in 40% of cases the highest IOP is found in the early morning and in 65% of cases it happens before noon. De Vivero et al. [74] observe the variation of IOP during the daytime of 101 untreated low-tension glaucoma (LTG) patients. They record IOP value every two hours and they observe a decreasing trend of IOP during the daytime. Hughes et al. [75] study the value of 24-hour IOP monitoring in routine clinical practice by observing 29 glaucomatous patients for 24 hours. They observe no difference between the mean clinic and the mean of the 24-hour monitoring,

however the peak of IOP during the 24-hour recording is 4.9 mmHg higher than the peak clinic. They suggest to consider the IOP fluctuations and spikes for the routine clinical evaluations.

As explained in [76], continuous IOP monitoring techniques are divided into two categories: (i) wired and (ii) wireless. Many wireless devices use active or passive inductively coupled telemetry. Active devices have a more complex design but are more robust and yield better communication performances. As a major drawback, they are usually implanted by substituting the native lens, therefore they require a surgical operation. As a consequence, this solution is usually adopted for patients who need to change the native lens, as those affected by cataract. Passive devices require a high coupling with an external antenna. If the coupling is too weak, then even a small change of the relative position between the device and the antenna can cause important differences between the real and the measured IOP.

From another point of view, continuously monitoring IOP can be effectuated: (i) permanently by using an implantable sensor, as explained in [77], or (ii) temporarily by using a contact lens sensor. As we are interested in wearable health monitoring systems, we focus on wireless non-implanted devices. Researchers and companies in the last years have developed smart lenses able to continuously measure IOP or other related parameters:

- Sensimed SA, a Swiss company in Lausanne [78] develops Triggerfish® [79], a contact lens able to capture spontaneous circumferential changes at the corneoscleral area. A micro-sensor is embedded in the contact lens and an antenna is placed around the eye. The antenna receives the information about the circumferential changes at the corneoscleral area and it is sent to a recorder device. Data can be read via Bluetooth on a computer by using their own software. More details will be provided below in the Section 4.3.

- Ziemer Ophthalmology [80] presents the DCT-lens [81], that is able to directly measure the IOP for 24 hours. The main advantage of this type of lens is that IOP is directly measured without capturing an indirect IOP-related physical quantity. An evaluation of a wired version of the contact lens has been provided in [82].

- Researchers of the ETH of Zurich [83] develop a thin-film transistors that can be easily inserted into everyday objects. It can be used for measuring several physiological signals and one among the possible applications is the IOP monitoring. This is realized by using the thin-film transistors along with strain gauges on a contact lenses. They test the contact lens with an artificial eye and they successfully measure the IOP. The development is still ongoing, given that for realizing a commercial device some problems have still to be solved, as the integration of the energy supply and the optimization of the electronics components placement.

- Yan [84] presents a microfluidic pressure sensor made of Polydimethylsiloxane (PDMS), a silicon-based organic polymer, and a colored dyed glycerol. The PDMS is used as structural element for the contact lens and the dyed glycerol is contained into a circular sensing chamber network inside the lens. The fluid displaces proportionally to the IOP and, by observing its displacement, the IOP can be determined. The main drawback is that the displacement has to be optically observed when eyelids are open. As a consequence, the device can not be used during the sleep phase.

- Chen et al. [85, 86] create a contact lens able to monitor IOP. They measure the change in corneal curvature by using an inductive coil embedded in the lens. They create a resonator (LC) by coupling the inductive coil (L) with a capacity (C). IOP variations cause corneal curvature variations that make the LC circuit resonating in a different way. They measure the resonance frequency of the LC wirelessly. They observe an excellent linearity between the IOP changes and the resonance frequency changes when using a silicon model eye.

- Cong et al. [87] realize a miniaturized capacity pressure sensors on PDMS material. By using a thermal compression, the sensor can take the desired form, as a contact lens. By using a metallic sphere, they test a contact lens containing the capacitive pressure sensors. The capacitive pressure sensors are disposed as an array, able to sense any pressure distribution. However, the read-out circuitry is not presented.

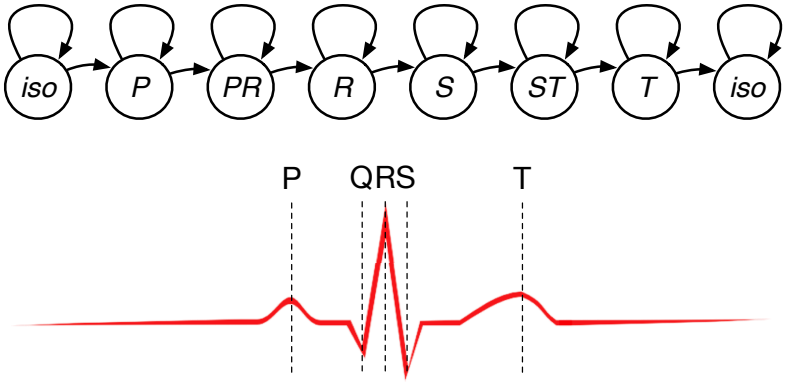


Figure 4.4: An example of HMM applied to an ECG signal. Every hidden state is a specific interval or a waveform of the ECG, as the P-wave, PR interval, QRS complex, ST interval, T-wave, U-wave. The left-right topology is used for representing the sequential activity of the cardiac cycle.

IOP-related monitoring data has been used by ophthalmologist for treating patients. Mansouri and Shaaraw [88] show how this information is important for the decision of treatment. They study 15 glaucomatous patients by comparing the standard IOP measurements and continuous monitoring for 24-hours using the Triggerfish® of Sensimed SA. 69% of the patients present the IOP peak during their sleep period and 80% have prolonged IOP peaks. The usefulness of the continuous IOP-related monitoring data has been shown in practice, because for 73% of the patients the ophthalmologists change the treatment, e.g. changing the timing of a treatment, adding anti-glaucomatous drops or proposing alternative treatments as laser and surgery.

4.2.5 State-based modeling on physiological signals

Physiological signals can be thought of as time series where the physiological parameter changes over time. Often these signals are characterized by the repetition of certain patterns due to the cyclic nature of the human biological system. Several cycles are short term and semi-automatic due to our physiology, e.g. the cardiac cycle and the respiratory cycle. Longer term cycles are more behavioral, such as the *circadian rhythm*. This term is derived from the latin *circa dies*, that means "approximately a day". The term indicates the repetition of human behaviors and physiological patterns that occur within a 24 hour period. The circadian system is regulated by endogenous clock genes and the suprachiasmatic nucleus in our brain is responsible of the synchronization of the peripheral tissues located for instance in the eye, brain and heart. The light of day and the darkness at night is the principal factor for the regulation of our circadian rhythm [89]. State-based modeling can naturally describe the repetition of sequences of patterns in time, for this reason it is widely used in this context.

The cardiac cycle is described by the alternation of the diastole and the systole, corresponding respectively to the relaxation and contraction of the heart. Several works have tried to capture this cyclic nature by using state-based modeling. In particular HMMs are popular in the analysis of the ECG signal. Coast et al. [90] use HMMs with a left-right topology for representing the sequential activity of the cardiac cycle, as shown in Figure 4.4. Every hidden state is a specific interval or a waveforms of the ECG, e.g. P-wave, PR interval, QRS complex, ST interval, T-wave, U-wave. By using the Viterbi algorithm, the model is able to recover the sequences of waves and intervals. In addition, they build a HMM that includes a normal, a supraventricular and a ventricular cycle. They test their model for the detection of the arrhythmia, obtaining similar results as the ones of commercial systems. Moreover, they find an accurate detection of the P-wave occurrences and locations. Andreao et al. [91] present a different approach using HMMs for modeling the ECG signal. They move from the left-right topology to another more complex but able to model a larger variety of beats. HMMs

are trained on different persons for making their system able to be used on automatic ambulatory analysis. The ECG signals are used for feeding HMMs without being manually labeled. The models are tested on public databases and they obtain favorable results when compared to other works in the literature. Koski [92] uses a HMM without injecting any knowledge about the hidden states. He tests several topologies having a greater number of states than the previous works in the literature. He finds that 20 states are enough for modeling the ECG signals. He analyzes the trained model and searches for the correspondences with the ECG structure. He trains two models: the first one is trained with normal ECG and the second one with ECG presenting premature ventricular beats. By using the models previously trained, he performs the classification task for the recognition of pathological signals. The analysis of the ECG signal with state base modeling is used also for the detection of particular diseases, as the apnea-bradycardia [93].

The electromyography (EMG) records the electrical activity produced by skeletal muscles. Several efforts have been done in the EMG-based speech recognition, gesture recognition and activity recognition fields. Chan et al. [94] perform automatic speech recognition by using EMG signals from five articulatory muscles. They compare two machine learning algorithms, i.e. linear discriminant analysis (LDA) and HMMs on a vocabulary of ten words. They find HMMs resilient to temporal variance, in fact they do not require the temporal alignment of the data, unlike LDA. Even if in their case LDA performs better than HMMs, they conclude that HMMs would perform well in a multimodal speech recognition system. Other works deal with this problem using different machine learning techniques, as multi-stream HMMs [95]. Meng et al. [96] recognize the gait phase by using HMMs on EMG data. They use a model where hidden states correspond to different phases of the gait: Early stance, Mid stance, Pressing, Swing flexion and Swing extension. They record the EMG activity of four muscles of the leg while walking on a treadmill and compute the features by windowing the signal. They try different set of features and windows lengths, achieving, in their best case, an accuracy rate of 91.9%. Caon et al. [97] are using HMMs for activity recognition with the EMG signal. Five activities are recognized: Walking, Running, Cycling, Sitting and Standing. Four muscles are recorded from eight subjects and the activities are modeled by using two hidden states. They compare two systems: the first one is trained on several users' data and the second one is specialized on single subject data. The comparison is made by using a 10-fold cross validation and the leave-one-subject-out technique and, as expected, the system provides better results in the first case. Chiang et al. [98] propose a multivariate-autoregressive HMMs (HMM-mAR) to model muscle activities during movements. The connection between muscles is represented by using a graph network. The effectiveness of their work is illustrated through a classification of Healthy and Stroke subjects in the context of hand gesture recognition. As result, the classification accuracy rate increases when increasing the number of features. Only three features provide good results, i.e. an error rate less than 10% on this two-classes problem.

The EEG records the electrical activity of the brain. This signal is generated by the current flowing through the neurons. The amplitude of the EEG is usually very small, typically it ranges between 10 and 100 μV, and the signal can hardly be distinguished to the background noise. A very common technique for avoiding this problem consists in the use of the evoked potentials, i.e. the computation of the average of the EEG activity after that an external stimulus is repetitively presented to the subject. Given the difficulty of the task, several different machine learning algorithms are currently used in the literature without a leading technique [99]. Obermaier et al. [100] perform the classification of EEG signal while the subject is imagining to perform left or right hand movements. They use a left-right HMM with a variable number of states and mixtures per state. They compare HMMs with a linear discriminant (LD) classifier. Online and offline classifications are compared and in both cases HMMs perform better than LD. In particular, HMMs is trustworthy in the online analysis. Zhong and Ghosh [101] face the problem of the multi-channel EEG classification by using state-base modeling. They use two databases containing two groups of subjects: Alcoholic and Control. Their aim is to determine to which group a person belongs. They compare the results obtained by using several techniques including HMMs, Coupled HMMs (CHMMs), Factorial HMMs (FHMMs) and distance coupled HMMs (DCHMMs). They use HMMs with 5 hidden states for modeling the EEG signals. They found that the simple multivariate HMMs perform better than the other methods, achieving an average accuracy rate of 90.5% for the first database and 78.5% for the second. Chiappa and Bengio [102] aim to separate three mental tasks from the EEG signal. The subject concentrates on a mental task for the desired time and after he changes to another mental task without resting between

the two. Two subjects participate to the tests for three consecutive days. They compare four machine learning techniques, i.e. GMMs, HMMs, input-output HMMs (IOHMMs) and MLP, on segments of 1, 2 and 3 seconds extracted from the recordings. They find that larger time windows provide better results and the discriminative modeling, i.e. IOHMM and MLP, performs better. Lederman and Tabrikian [103] use parallel HMMs (PHMMs) for the classification of multi-channel EEG signals. They use an artificial EEG database and two real databases concerning the hand movements. They firstly segment the signal and compute a set of temporal and spectral features for each segment. Therefore they apply PHMMs with 5 states and 3 mixtures per state and compare the results with other machine learning techniques, as Bayes classifier, LDA, and ANN. PHMMs outperform the other classifiers achieving an improvement in the accuracy rate of 9% and 2% on the two datasets.

The respiratory cycle is described by the alternation of inhalation and exhalation, permitting the exchange of oxygen and carbon dioxide between the lungs and the environment. Al-Ani et al. [104] propose to use HMMs for the identification of sleep disorders. They analyzed the polysomnography signals, that monitor many functions at the same time, as EEG, EMG, ECG and the respiratory airflow. The Viterbi algorithm permits to find the pathophysiological state sequence, for instance Snoring, Inspiration, Expiration, Obstructive apnea, etc. They perform some tests on real data, obtaining accuracy rates between 90% and 100%. Ravishankar et al. [105] analyze data relative to the respiration rate and the pulse-oximetry for detecting respiratory distresses by using Markov models. They focus on patients suffering from respiratory complications. Their algorithm is able to detect the phase in which the respiratory complication begin, while existing systems would not alarm. They obtain a true positive rate of 92% and a false positive rate of 6%.

In conclusion, generative modeling appears widely used in the context of physiological signals. In particular, signals as the electrocardiographic (ECG), electromyographic (EMG) and electroencephalographic (EEG) appear commonly analyzed with generative modeling as HMMs. The majority of the works goes in two directions: (i) a 2-classes classification problem, mostly health vs illness, and (ii) N-classes classification problem, usually for the detection of particular conditions. Apparently HMMs are widely used and, in our opinion, this is probably due to the state-based nature of the signals that cyclically repeat typical patterns. Finally, we have noticed that often complex versions of HMMs are used, as CHMMs, FHMMs or DCHMMs, probably to better cope the specific characteristics of certain physiological signals.

4.3 Sensimed smart lens

In this section we will present our work using data coming from the smart lens produced by Sensimed SA. As specified in the Introduction, this work is related to the project Sensimed Data Intelligence, elaborated during this thesis.

4.3.1 Introduction

Sensimed SA has developed a wearable contact lens sensor (CLS), called Triggerfish®. This medical device is a highly oxygen-permeable soft CLS, having 2 active sensing-resistive strain gauges, capable of recording circumferential changes in the area of the corneoscleral junction of the eye. The output of the sensor is expressed in electric arbitrary units, i.e. millivolt equivalent (mVeq). A strong correlation between the IOP and mVeq has been shown [106]. An example of the Sensimed lens is shown in Figure 4.5a and Figure 4.5b.

The CLS device is safe and non-invasive. A healthcare professional applies and removes the CLS on the patient. The device is usually worn for 24-hours while the patient can perform its usual activities. A micro-sensor is embedded in the CLS and it communicates with an antenna placed around the eye, as shown in Figure 4.5c. The communication between the micro-sensor and the antenna is wireless. Data is recorded on a holder device carried by the patients. At the end of the recording, the healthcare professional downloads the data via Bluetooth and is able to visualize the data by using a dedicated software. Finally, the data is anonymously and automatically sent to the servers of Sensimed.

The CLS is able to record continuous ocular dimensional changes up to 24 hours. Every 5 minutes an acquisition of 30 seconds at the sampling frequency of 10 Hz is recorded, generating sequences of 300 points, called *bursts*. Bursts can appear very different depending on the awake/sleep status

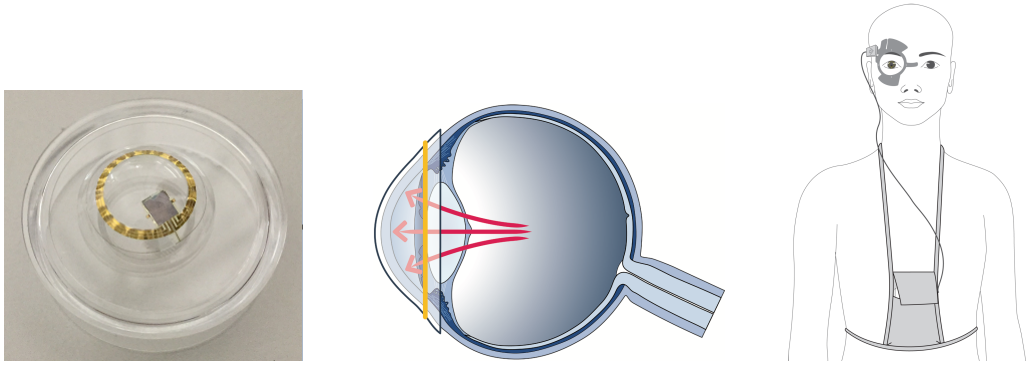


Figure 4.5: Representation of a real Sensimed smart lens (a), when positioned on the eyeball (b) and the equipment provided by Sensimed SA for the recording of the smart lens data (c). The images (b) and (c) are taken from [78] with the authorization of the authors.

of the patient. During the daytime, the patient is generally awake and she or he blinks his eyes. Blinks produce peaks of pressure in the CLS recordings. An example is shown in Figure 4.6. During the nighttime, the opposite occurs: when the patient is sleeping, she or he does not blink the eyes, therefore finer details, as the pulse of the hearth rate, are visible, as shown in Figure 4.7. The scale of the y-axis in the two figures is different. More details about the signals will be provided in the next section. In order to observe the 24-hours shape of the CLS recordings, the so-called *profile* is computed. The profile is a multivariate time series containing the sequence of features computed from each burst. An example of a profile is given in Figure 4.8.

4.3.2 Database

In this chapter we present our work consisting in the discrimination between healthy and glaucomatous patients. Through different clinical studies around the world, Sensimed has created a database containing several CLS recordings. Meta-information of the patients is also available, as health status, age, IOP measurement, etc. All information is completely anonymized for privacy protection.

As said in Section 4.2.4, several types of glaucoma exist. The type of glaucoma is suspected to influence the CLS recording. For this reason, we decide to not group all the glaucomatous patients but we separate them depending on the specific type of glaucoma. Given that our aim is the discrimination between healthy and glaucomatous patients, we choose the POAG type, which is the most common form of glaucoma and also the most recurrent type in the Sensimed database.

We later apply a filter selecting the CLS recordings having a total duration of the recording longer than 20 hours. This condition assures that both the awake and sleep periods are present, as they are suspected to contain useful information for the discrimination task. After the filter phase, we obtain two unbalanced sets, containing 271 Healthy and 374 POAG profiles.

4.4 Feature extraction

Referring to Figure 2.1 of Section 2, the feature extraction should retrieve informative and not redundant characteristics from initial data. Often the feature extraction applies complex mathematical operations on the input data, completely transforming the initial characteristics. As a consequence, the set of generated features is sometimes difficult to be interpreted. In the medical context, this can represent a problem as it is useful for the healthcare professionals to interpret all the characteristics and relate them to physiological mechanisms. In our case, all the features extracted have a physiological meaning or are related to a physiological functioning.

In our work, we use an *intra-burst* approach. Coherently with the notation previously introduced, we note the time series of features as $\mathbf{X} = \{\mathbf{x}_n : n \in N\}$, where N is the total length of the time series. Considering a multivariate time series $\mathbf{X}$ having D dimensions (number of features), the n-th feature $\mathbf{x}_n$ is equal to $\{x_{1n}, \ldots, x_{dn}, \ldots, x_{Dn}\}$. The set of features is extracted from each burst,

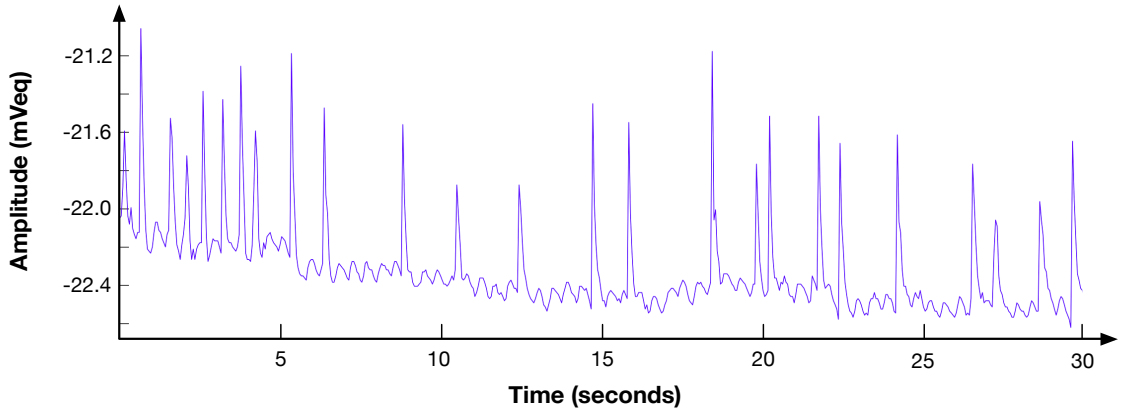


Figure 4.6: An example of a burst recorded during the awake period. The blinks introduce peaks in the CLS recordings, namely 25 in this example. Between the blinks the ocular pulse is visible.

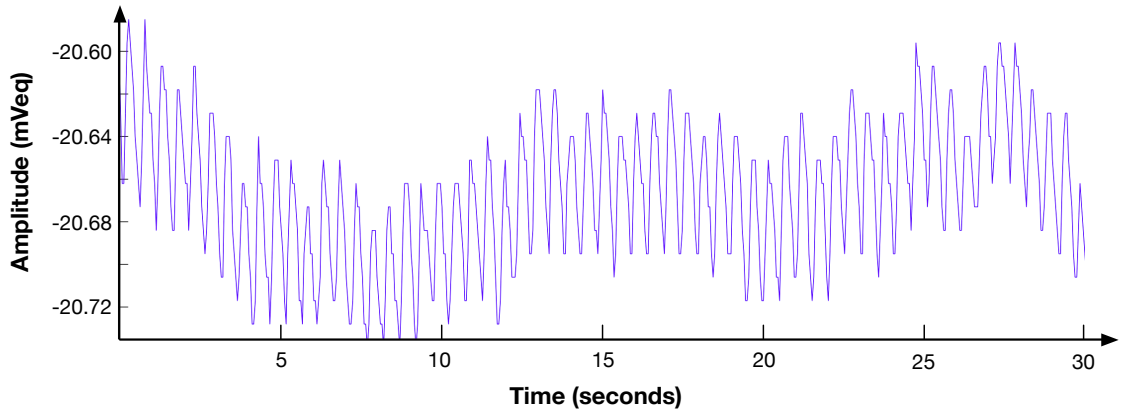


Figure 4.7: An example of a burst recorded during the sleep period. A burst contains 300 values recorded during 30 seconds by the lens. The ocular pulse is clearly visible given the absence of blinks in the recording.

constituting a feature sample $\mathbf{x}_n$ with n representing the burst index. Given that bursts are equally spaced in time, we can concatenate the sequence of feature samples in order to generate a time series $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N\}$ with $\mathbf{x}_n$ the feature vector extracted from the burst at the time n. A single burst can be though as a multivariate time series, $\mathbf{B} = \{\mathbf{b}_j : j \in L_b\}$, where L_b is the total length of the burst. The sample $\mathbf{b}_j$ has two dimensions in which we are interested: (i) the circumferential changes in the area of the corneoscleral junction of the eye and (ii) the power of the device.

We compute three types of features:

- **Statistical features** are those computed in the temporal domain by applying simple statistical methods. Typically for every burst we compute the *amplitude*, the *minimum*, the *maximum*, the *standard deviation*, the *skewness* and the *kurtosis* using the circumferential changes in the area of the corneoscleral junction of the eye. The skewness gives an indication on how far data is shifted to the right or left of the normal curve. It is computed as:

$$skewness(n) = \frac{1}{L_b} \sum_{j=1}^{L_b} \left(\frac{\mathbf{b}_j - \mu_n}{\sigma_n}\right)^3 \tag{4.1}$$

  where μ_n and σ_n are respectively the mean and standard deviation of the n-th burst.

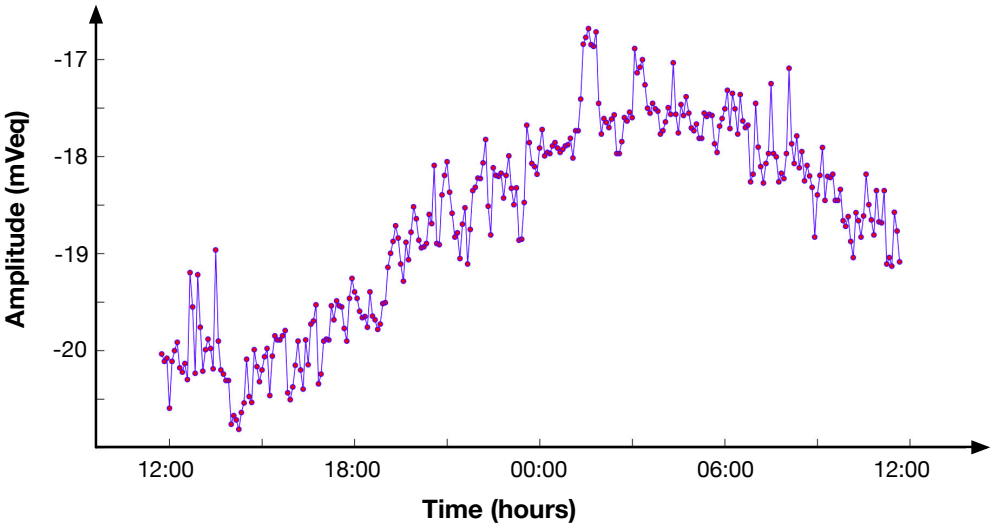


Figure 4.8: An example of a profile. In this case, each point in the time series is the median of the mVeq of the corresponding burst. For this representation, the median has been chosen instead of the mean for being more resilient to intra-burst outliers.

The kurtosis gives an indication on the peakedness or flatness of data relative to a normal distribution. It is computed as:

$$kurtosis(n) = \frac{1}{L_b} \sum_{j=1}^{L_b} \left(\frac{\mathbf{b}_j - \mu_n}{\sigma_n}\right)^4 \tag{4.2}$$

- **Physiological features** are the features that have a physiological meaning. We compute two different features: (i) the *Ocular Pulse Amplitude* (OPA) and (ii) the *Blink density*, both using the circumferential changes in the area of the corneoscleral junction of the eye. The first feature is computed by analyzing portions of the burst where the hearth rate effect is visible. More information will be given below with a description of our procedure to compute OPA. The blink density is typically defined as the number of blinks made per unit time, however in our case we will refer to the duration of a bursts.
- **Device-based features** are the features coming by the measuring device. In this case we include the *Pwmstd*, that is the standard deviation of the power of the device. This feature is suspected to be related to eye movements.

4.4.1 Ocular Pulse

An extensive definition of the ocular pulse has been reported in [107]:

The ocular pulse is the pulsatile pressure curve obtained by continuously recording intraocular pressure. During systole, the ocular and orbital vascular beds are filled. Since the scleral coats resist distention, intraocular pressure rises. Short circuiting the arterial run-off (as occurs systemically with aortic insufficiency) results in a widened pulse pressure curve. Similarly, the pressure pulse curve is widened in carotid cavernous sinus fistula. Since the short circuit runs into the venous tree, elevation in venous pressure occurs, and there is distention of the conjunctival and episcleral veins, and an increase in intraocular pressure.

Detailed information about the ocular anatomy can be found in [108]. The ocular pulse amplitude

(OPA) is expected to be particularly interesting for the classification task. In particular, some papers in literature indicate the OPA to be related with certain types of glaucoma, as in [109, 110, 111]. Other works find a relation between IOP and OPA, as in [112].

We describe now our procedure to compute the OPA from the intra burst signal. As previously said in the Section 4.3.1, the CLS signal is influenced by the eye blinking that produces important peaks in the signal. When the blink is not present, typically during the sleep period and between two blinks, the ocular pulse is visible. The peaks introduced by the eye blinking are several times greater then the ocular pulse and they cover up the ocular pulse in the CLS signal. Physiological information as the frequency of the ocular pulse and its amplitude can then be retrieved from portions of the signal not affected by eye blinking. The ocular pulse depends on the systolic/diastolic cycle, that is related to the heart rate pulsation.

For finding the portions of the CLS signal not affected by the eye blinking, we use a sliding window approach with overlaps. By analyzing the content of the sliding window we are able to extract the ocular pulse information if present. We choose a sliding window of 5 seconds with an overlap of 4.5 seconds. The length of the sliding window is a compromise between the quantity of windows and reliability. A short window has more probability to find ocular pulse portions, however the computed parameters are less precise in both time and frequency domain. The choice of the length of the overlap is a compromise between quantity of windows and computational load. A long overlap ensures a large number of windows that however have systematically to be analyzed. Each window is analyzed for finding the portions of the CLS signal containing the ocular pulse. We effectuate an analysis in the frequency domain and we zero-padded the windows in order to increase the frequency resolution. The zero-padding technique consists in adding a certain number of zeros at the end of the temporal signal [113]. We compute the Discrete Fast Fourier Transform (DFFT) and we filter out the zero-frequency spectral component.

As a first step, we search for a peak F_{peak} in the frequency domain, in order to retrieve the ocular pulse frequency (OPF). We search the peak between $F_{low} = 0.7$ and $F_{high} = 1.85$ Hz, that respectively correspond to 42 and 110 beats per minute (bpm). Similar techniques are used on similar signals for finding the heart rate frequency, as the photoplethysmographic signal [114]. We discard the windows in which the F_{peak} is equal to F_{low} or F_{high}. Given that the sampling frequency F_S of the Triggerfish® lens is 10 Hz, the frequency band is below 5 Hz according to the Shannon theorem.

For every window we compute a quality factor QF in order to have an information about the reliability of the measure. QF should be high when the ocular pulse is clearly identifiable and it has to be low when blinks cover up the ocular pulse. QF is proportional to the ratio between the energy of the signal in the frequency band containing the information (around the F_{peak}) and the rest of the band. The frequency band around the F_{peak} is computed between $F_{peak} - F_{BD}$ and $F_{peak} + F_{BD}$, where F_{BD} has been set equal to 0.2 Hz after some tests. The rest of the band ranges between 0 and $F_{peak} - F_{BD}$ Hz and between $F_{peak} + F_{BD}$ and 5 Hz. An example is shown in Figure 4.9: the green lines represent the frequency band around the F_{peak}, while the purple lines represent the frequency band where we search the F_{peak}. The QF is calculated by the following Formula:

$$QF = \frac{\sum_{f=F_{peak}-F_{BD}}^{F_{peak}+F_{BD}} Y_f}{\left(\sum_{f=0}^{F_{peak}-F_{BD}} Y_f + \sum_{f=F_{peak}+F_{BD}}^{F_S/2} Y_f \right)^2} \tag{4.3}$$

where Y_f is the frequency band. At the nominator we compute the energy of the signal in the frequency band containing the information, while at the denominator there is the rest of the frequency band. As expected, QF attains high value during the sleep period, while the opposite occurs during the awake period. We compare QF with a threshold value, called QF_{thr}. If QF is greater than the threshold, therefore the window is accepted, otherwise it is rejected. Given the burst duration (30 seconds), the window length (5 seconds) and the overlap (4.5 seconds), a total of 55 windows are computed for each bursts. If at least 3 windows have a QF value greater than QF_{thr}, we compute the representative value $OPF(n)$ for the burst as weighted mean of the single F_{peak} values, computed as

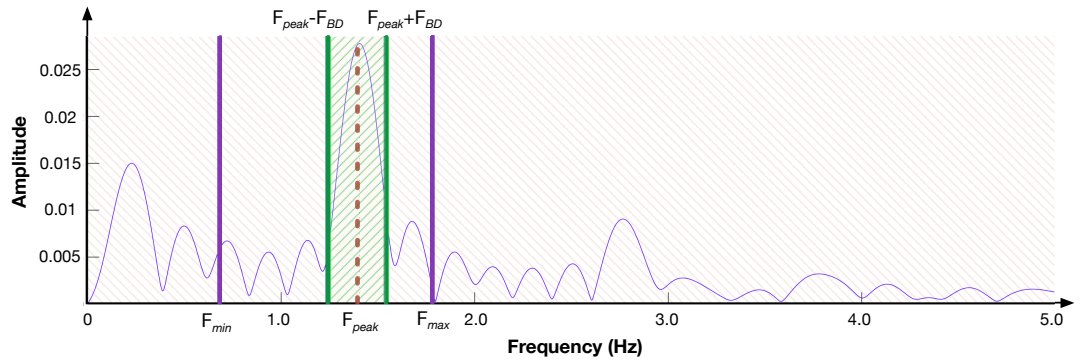


Figure 4.9: Ocular pulse in the frequency domain. The purple lines represent the frequency band where we search the frequency peak, called F_{peak} and depicted with a red dotted line. When F_{peak} is found, a frequency band around the F_{peak} is considered, i.e. from $F_{peak} - F_{BD}$ to $F_{peak} + F_{BD}$. This frequency band is represented by the green lines and is used to compute the QF.

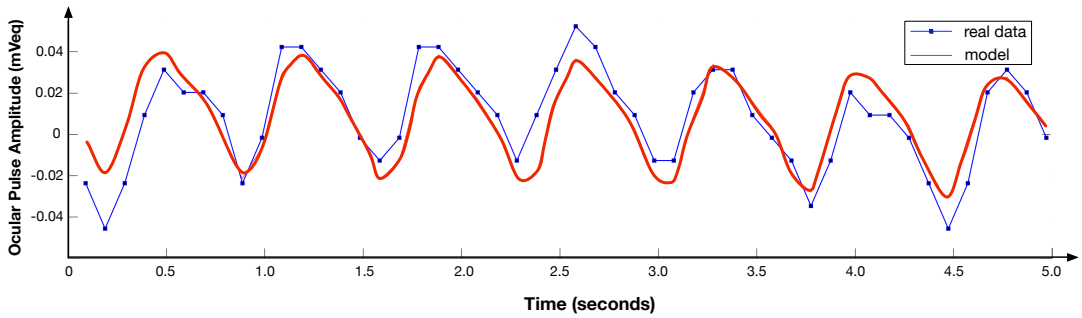


Figure 4.10: Ocular pulse in time domain. We represent an example of the real ocular pulse (blue line) and the estimated model (red line). The estimated model is used for computing the ocular pulse amplitude.

follows:

$$OPF(n) = \frac{\sum_{w=1}^{W} F_{peak,w} QF_w}{\sum_{w=1}^{W} QF_w} \tag{4.4}$$

where W is the total number of windows computed in the burst. The heart rate in bpm is computed by simply multiplying the $OPF(n)$ per 60.

For computing the OPA, we use a model able to fit the data and representing the ocular pulse shape. We notice that the form of the ocular pulsation is very similar to the PPG signal. We create a simple model taking as input the F_{peak} and able to model a constant, a linear drift, the first and second harmonic of the frequency peak. This model permits to reduce the computational load, while easily computing the OPA. The model is represented by the following equation:

$$y = a + b\ x + c\ sin(2\pi F_{peak} x + d) + e\ sin(4\pi F_{peak} x + f) \tag{4.5}$$

where a, b, c, d, e and f are the parameters of the model to be estimated. For every window having QF greater than QF_{thr} we fit the model parameters and we deduce the model equation without drift:

$$y' = \hat{c}\ sin(2\pi F_{peak} x + \hat{d}) + \hat{e}\ sin(4\pi F_{peak} x + \hat{f}) \tag{4.6}$$

We compute the OPA as the difference between the maximum and the minimum of y'. As for the OPF, if at least 3 windows have a QF value greater than QF_{thr}, we compute the representative value

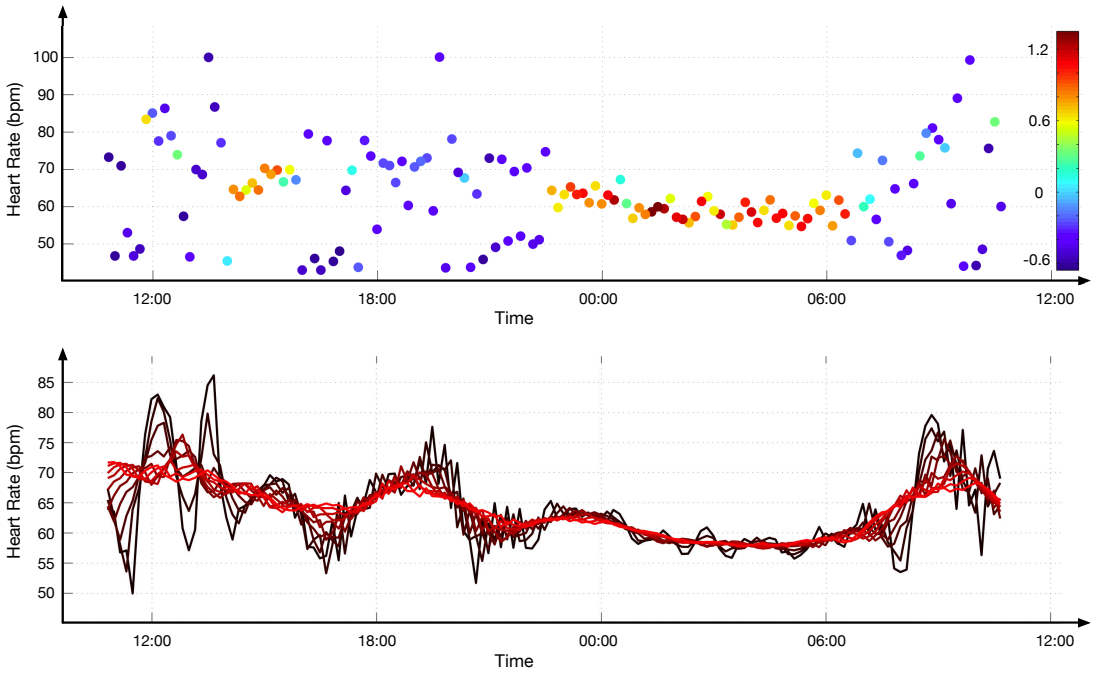


Figure 4.11: Example of a time series of the Ocular Pulse Frequency (OPF). In (a) we plot the points computed using a scale of colors proportional to the $log(QF)$. Points in the range of yellow to red colors have a better quality factor than the ones in the dark blue range. In (b) several smoothed functions are applied to the scatter points using a black/red scale.

$OPA(n)$ for the burst as weighted mean of the single OPA values, computed as follows:

$$OPA(n) = \frac{\sum_{w=1}^{W} OPA(n)_w QF_w}{\sum_{w=1}^{W} QF_w} \tag{4.7}$$

where W is the total number of windows computed in the burst. In Figure 4.10 the blue line represents the real ocular pulse, while the red line represents the estimated model.

We try different values of QF_{thr} and, after some tests, we decide to take $QF_{thr} = 0.2$. This value is quite low, however it permits to calculate the OPA for at least 90% of the bursts. In Figure 4.11 and 4.12 we report an example of the OPF and OPA time series extracted from a specific CLS recording. The recording lasts 24 hours, from 11 AM to 11 AM of the next day. The patient reports to have slept after lunch. In the upper part of the two figures we report the evolution of the OPA and OPF as scatter points. We represent each point using a scale of colors proportional to the $log(QF)$. We observe that the sleep period is easy detectable for two reasons: the stability of the signals and the better quality of the data. In the bottom part of the figure, we apply several smoothing functions to the scatter points. 10 smooth functions using analysis windows of different lengths have been applied, i.e. from 3 to 21 points with a step of 2 points. More the points in the analysis window, the smoother the curve. In the figures the curves are represented in a black/red scale proportionally to the length of the analysis window. We observe that the OPF and the OPA have a great variability during the awake period, compared to the sleep period. However, it should also be considered that the quality of the points is in average lower during the awake period and consequently the values are less reliable.

4.4.2 Blinks

Another physiological feature that we have computed is the *blink density*, namely the number of times a person blinks in a certain amount of time. Information on blinks could potentially be related to the glaucoma disease. Johnstone et al. [115] relate the blink-related IOP elevations with the aqueous

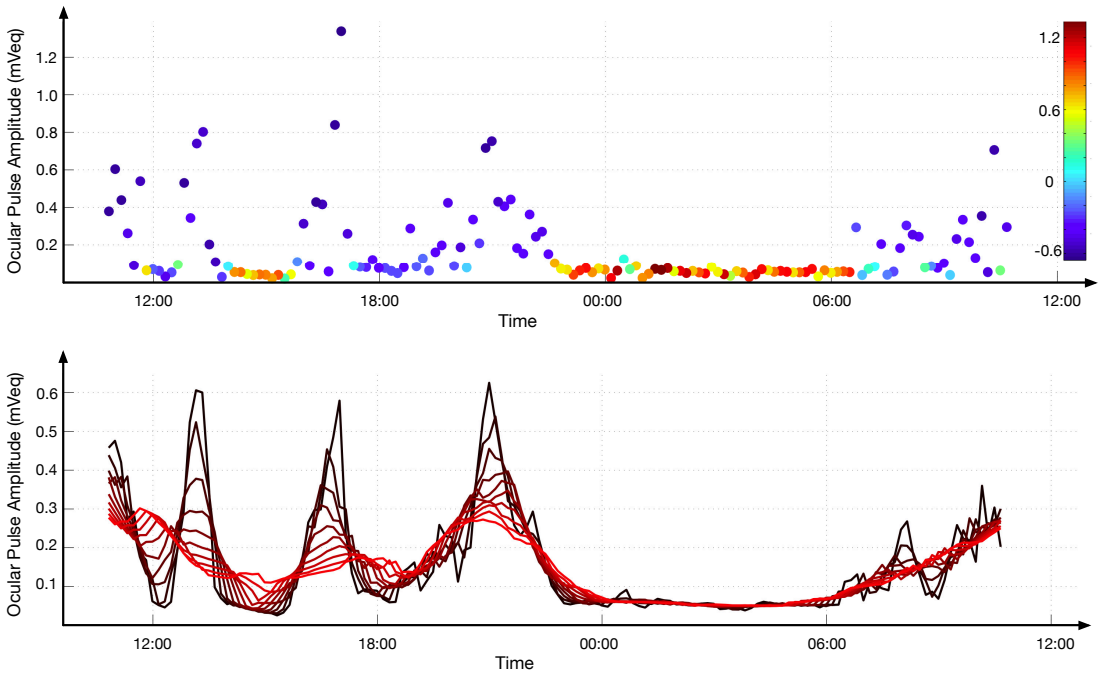


Figure 4.12: Example of a time series of Ocular Pulse Amplitude (OPA). In (a) we plot the points computed using a scale of colors proportional to their quality computed through the $log(QF)$. In (b) several smoothed functions have been applied to the scatter points using a black/red scale.

humor outflow. They observe the presence of a pulsatile aqueous outflow mechanism with a pulse wave source originating from Schlemm's canal. In particular they model the process by asserting that a blink functions as a source of force, Schlemm's canal as a reservoir, trabecular meshwork as a compressible tissue, and collapsible tubes or pores at the level of Schlemm's canal acting as one-way valves. In another work, they show that the optic nerve head undergoes motion in response to cardiac pulse induced transients [116]. As a consequence, excursions of the optic nerve head could be considerably larger with blinks and eye movements, which cause larger amplitude pulse transients.

As previously stated, the consequence of eye blinking is the introduction of a peak in the CLS signal, covering the ocular pulse. In order to detect the blinks, we have developed an algorithm able to retrieve and evaluate the peaks in the CLS signal [117]. In a given burst, we check all the points in order to find blinks. A peak is characterized by the sequence of three successive and continuous local extrema: (i) a local minimum (the start point), (ii) a local maximum (the peak point), and (iii) a local minimum again (the end point). As a first step, we identify all the peaks, represented by the points having the sequences of nearby points toward the left and right strictly monotonically decreasing until reaching a local minimum, namely the blink start and the end points. In the next step, in order to consider a peak as a blink, we check if the difference between the peak and the start point and the difference between the peak and the end point are greater than a threshold (T_{blink}). We choose the value of T_{blink} proportional to the mean amplitude of all bursts in a given profile. By checking all the peaks, we are able to find the list of blinks. In Figure 4.13 we show a synthetic example.

The number of blinks is variable during the awake period, because they depend on the activity performed by the patients. For example some activities as reading or watching the television are related to a lower frequency of eye blinking. As expected, the number of blinks during the sleep period is almost zero. Sometimes it can happen that variations in the CLS recording verified during the sleep period are wrongly interpreted as blinks. These abnormal variations are suspected to be related with the Rapid Eye Movement (REM) phase.

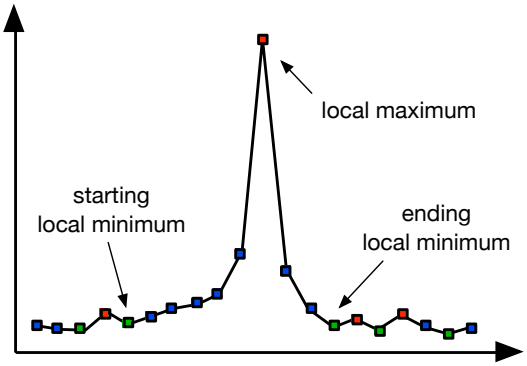


Figure 4.13: Synthetic example showing the functioning of our algorithm for the detection of blinks. As first step, the local maxima are found (red points). For every local maxima, we find the closest starting and ending points (green points). We compute the difference between the local maxima and the starting point and the difference between the local maxima and the ending point. If these differences are greater than the threshold T_{blink}, therefore the local maxima is considered as a blink.

4.4.3 Data pre-processing

The pre-processing consists in four phases: (i) **outlier removal**, (ii) **missing data replacement**, (iii) **dynamic coefficients** computation and (iv) **normalization**.

The detection of the outliers has been performed by using the Hampel method [118]. Hampel uses a measure of the robustness of an estimator against the outliers. He introduces the concept of "breakdown point", defined as the smallest percentage of outlier that can cause an estimator to take arbitrary large values. As a consequence, the larger the breakdown point, the more robust the estimator. The Hampel filter has been often described as extremely effective in practice for the outlier detection [119].

In practice, this method uses the median absolute deviation (MAD). Given a set of points $X = \{x_1, \ldots, x_i, \ldots, x_N\}$, where N is the total number of points, the Hampel method consists in the following steps [120, 121]:

1. compute the median value using all data $x_{med} = Med(X)$;
2. compute the deviation of each point to the median value $dx_i = x_i - x_{med}$;
3. compute the median value of the deviations $dx_{med} = Med(\{dx_1, \ldots, dx_i, \ldots, dx_N\})$;
4. compute $MAD = b\ dx_{med}$, where usually $b = 1.4826$, a constant related to the assumption of normality of the data;
5. then, we define an acceptance criterion:

$$x_{med} - 3 * MAD < x_i < x_{med} + 3 * MAD \tag{4.8}$$

   If the point is not accepted, then it is considered an outlier.

We apply the Hampel method to all the features and we remove the points detected as outliers.

As next step, we interpolate the features in order to recover the missing points. A point can be missing for two reasons: it has been impossible to compute the value, for instance when too few points are present in the burst, or the point has been removed in the previous step because it has been detected as an outlier. Independently of the reason, we interpolate the features in order to replace the missing points with the interpolated values. We use a spline interpolation, based on a cubic interpolation of the values at neighboring grid points.

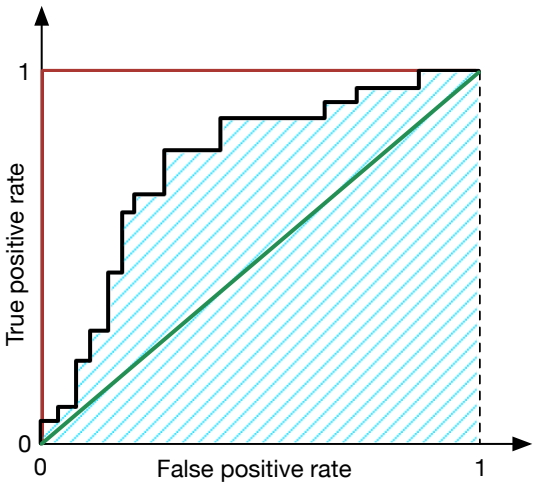


Figure 4.14: Synthetic example of three ROC curves. The red curve represents the perfect classifier, the green curve represents the random classifier, while the black curve represent an intermediate case. The AUC of the last ROC curve has been highlighted, i.e. about 0.75%.

Following the same procedure shown in the previous chapter, we compute the dynamic coefficients in order to include the information about the evolution of the time series. As shown in Section 3.4.1, we compute the velocity and acceleration coefficients, increasing the feature dimensional space of 3 times. Finally, we normalize data, by using the z-score normalization, the same procedure shown in Section 3.4.2. The means and the standard deviations are computed using all the feature vectors extracted from the bursts in the training set.

4.5 Classification

4.5.1 Protocol of evaluation

As first step, we need to define a metric for evaluating and comparing the performances of the machine learning algorithms. The machine learning community and most importantly the medical community often uses the area under the Receiver Operating Characteristic (ROC), called also Area Under Curve (AUC) statistic for model comparison [122]. The ROC curve is a graphical representation of the performances of a binary classifier when changing a discrimination threshold. The curve is obtained by plotting the false positive rate on the x-axis and the true positive rate on the y-axis. For both axes the scale varies between 0 and 1. The points of the curve correspond to a different threshold value. A completely binary random classifier, i.e. a system where decisions are taken by flipping a coin, would generate points along a diagonal line, also called line of no-discrimination. The more the curve approaches the upper-left corner, the better it is. The perfect classification consists on a curve going straight up the y-axis and then along the x-axis [123]. A synthetic example is represented in Figure 4.14.

The AUC is simply the integral of the ROC curve. The two-dimensional information represented by the ROC curve is summarized by a single scalar value. The completely binary random classifier has an AUC equal to 0.5, while the perfect classifier an AUC equal to 1. The AUC of a classifier is expected to range between 0.5 and 1. Generally speaking, classifiers are considered excellent when the AUC values range between 0.9 and 1, good for values between 0.8 and 0.9, fair for values between 0.7 and 0.8 and poor for values between 0.6 and 0.7.

We apply machine learning techniques for classifying the CLS recordings in two categories, Healthy and glaucomatous of type POAG. The data at our disposal consists in an unbalanced set, more precisely 271 Healthy and 374 POAG. Given that the data set is quite small and unbalanced, as

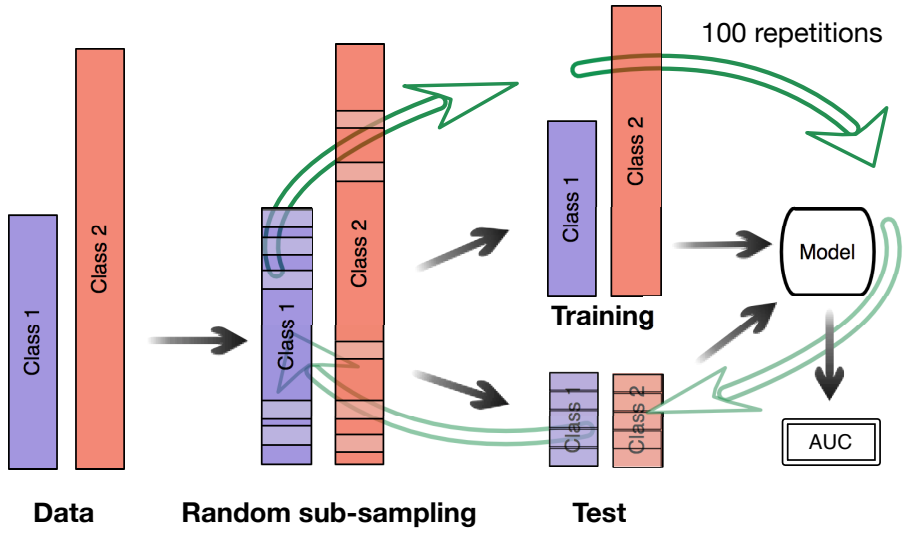


Figure 4.15: Procedure for evaluating the classification ability of the models. We use the *repeated random sub-sampling validation* with 100 repetitions. The test set is re-balanced by randomly selecting the 25% of the less represented class and an equal number of random elements of the other class. We use the AUC for comparing the classifiers.

evaluation method we use the *repeated random sub-sampling validation* with a balanced test set. The repeated random sub-sampling validation consists in randomly dividing several times the data set in two parts, i.e. training and test set. This evaluation method has been proposed and validated by Sensimed SA. The advantage of this procedure over k-fold cross validation is that the proportion of the train/test sets is not dependent on the number of iterations. The random separation is done by randomly selecting 25% of the less represented class, i.e. the Healthy, and selecting an equal number of random elements of the other class for having a balanced test set. The whole procedure is repeated a total of 100 times. We systematically compute the AUC for every repetition. Figure 4.15 illustrates this evaluation protocol. The results are reported in terms of AUC and with the 95% confidence interval, computed as follows:

$$C.I. = \mu_{AUC} \pm \frac{Z_{\alpha/2} \cdot \sigma_{AUC}}{\sqrt{N_r}} \tag{4.9}$$

where μ_{AUC} is the mean value of the AUC, σ_{AUC} is the standard deviation, N_r is the number of repetitions and $Z_{\alpha/2} = 1.96$ is the Z value for the desired confidence level at 95%.

4.5.2 GMMs/HMMs modeling

When applying a discriminative approach algorithm, we have to adapt the *protocol of evaluation* to the case. Given that the data set is unbalanced and the test set has been artificially balanced, the distribution of the two classes is different between training and test set. In practice, the test set has equal priors, while the priors of the training set are greater for the POAG class. As a consequence, it is easy to imagine that the less represented class has probably more difficulties in being correctly classified. In order to solve such problem, discriminative modeling requires the use of some operations that potentially modify the structure of the underlying information, as the repetition of the observations of the less represented class, the filtering of the observations of the more represented class or both the operations. While the first operation is potentially dangerous for the repetition of noisy observations, the second one is potentially dangerous for deleting some information useful for the classification task. For this reason, the first operation is generally preferred to the second one. While this situation is a problem for discriminative modeling, generative approaches as GMMs/HMMs do not bias the results in the favor of the more represented class, given that the priors can be modified

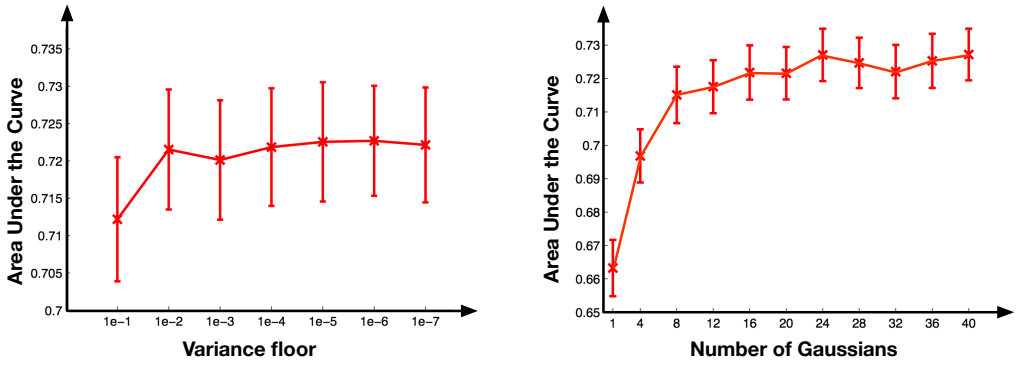


Figure 4.16: The trend of the AUC with the confidence interval when varying the variance floor (left) and the number of Gaussians (right).

on the fly.

Several parameters can be tuned in our GMMs/HMMs implementation:

- ν, the variance floor.
- K, the total number of Gaussians, representing the complexity of the models.
- N, the total number of states. As explained in Section 2.5.4, when $N = 1$ HMMs degenerate in GMMs, given that the density distribution of probabilities are parameterized by using an unique mixture of Gaussians. In addition, we test a specific topology by using the information about the sleep/awake status of the patient.

Variance flooring

When using a small set of data in the training set, the risk of over-training is high. When dealing with mixture of Gaussians as probability density estimator for GMMs and HMMs, the variance parameters are particularly susceptible of over-fitting. For instance few data points could generate a very small variance estimation that could not be representative of the underlying distribution of the data source. In order to solve this problem, the EM-algorithm can be modified imposing a lower bound on variance parameters, called *variance floor*. Any variance value going below the floor value during training is "clamped" to the floor value, representing a lower bound during the iterations of EM. However, the floor value has to be tuned [124].

In order to choose the best floor value, we compute the mean AUC value of several iterations by using a fixed number of Gaussians on a GMM. We choose $K = 30$ as a good compromise between complexity and execution time. In Figure 4.16 (left) we report the trend of the mean AUC when varying the variance floor value. With the exception of $\nu = 1e^{-1}$, the results are rather similar. The best value ($72.3 \pm 0.7\%$) has been obtained with $\nu = 1e^{-6}$, that we will use in further analysis.

Number of Gaussians

As next step, we observe the evolution of the accuracy rate when increasing the number of Gaussians K. For this test we set $\nu = 1e^{-6}$, $N = 1$. In Figure 4.16 (right) we illustrate the trend of the mean AUC when increasing K. We notice that the mean AUC tends to saturate around 20/30 Gaussians. We decide to keep $K = 24$ for the following tests, that yields an AUC of $72.7 \pm 0.8\%$

Topology

In this case, we try to take advantage of the state-based capabilities of HMMs. We use two strategies for determining the initial labeling of states: (i) linear alignment labeling and (ii) awake/sleep labeling.

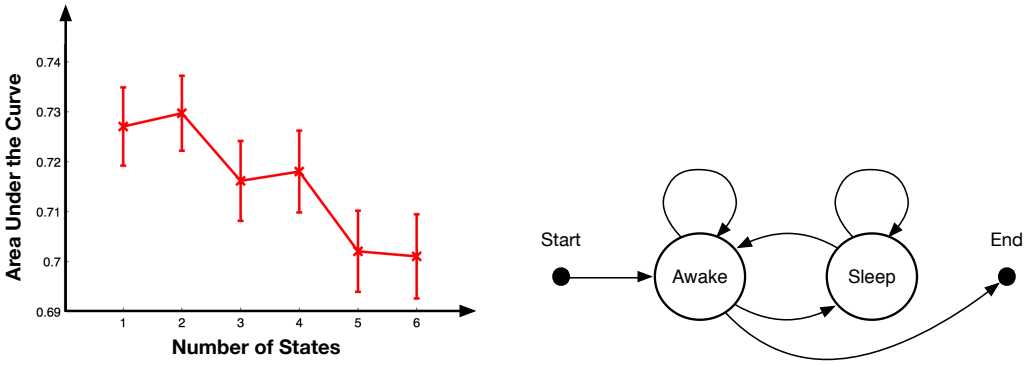


Figure 4.17: We use two strategies for determining the best topology: the linear alignment labeling and the awake/sleep labeling. In the left part, the trend of the AUC with the confidence interval when varying the number of states for the linear alignment labeling is reported, while in the right part the topology used with the awake/sleep labeling is reported.

In both cases, we use such information as starting point for the computation of the HMMs parameters. During the training phase, the HMMs iteratively find the most probable sequence of states that can converge to a different solution than the starting point.

- *Linear alignment labeling.* In this first approach we vary the number of states of the HMM from 1 to 6 while maintaining constant the complexity of the model, namely the total number of Gaussians. The Gaussians are equally distributed among all the states. A linear alignment labeling is computed over N_s segments, N_s being the total number of states, i.e. we linearly subdivide every profile in N_s portions and we associate these portions to the relative states.

- *Awake/sleep labeling.* In this second approach we use a specific 2-states HMM topology. The states represent the awake and sleep status of the patient. We use the information about the blinks in order to retrieve the labeling of the states.

In both cases, the labeling is used as initialization and the Viterbi algorithm finds the most probable sequence of states during the iterations. For both approaches we use $K = 24$ and $\nu = 1e^{-6}$. In Figure 4.17 (left) we report the trend of the mean AUC when increasing the number of states for the *linear alignment labeling* approach. We observe that the best AUC is obtained with the two states topology leading to $73.0 \pm 0.8\%$ AUC. However, since the difference with a simple GMM is not so relevant, we decided to continue the analysis by using the simplest model. In the Figure 4.17 (right) we show the topology of the awake/sleep model. In this case the topology has been adapted to the patients' profiles: according to the acquisition protocol, the patient is awake when inserting and removing the smart lens. This information is used by the specific topology, by adapting the probabilities of the non-emitting states. The model using the *awake/sleep labeling* does not seem to increase the performances of the classification. In fact, it provides an AUC of $72.3 \pm 0.8\%$, a slightly worse result when compared to the *linear alignment labeling* approach with 1 or 2 states.

Evaluation of the dynamic coefficients

We observe the evolution of the AUC when increasing the number of Gaussians for the three cases: (i) the original coefficient (x_n), (ii) the original and velocity coefficients ($x_n, \Delta x_n$) and (iii) the original, velocity and acceleration coefficients ($x_n, \Delta x_n, \Delta\Delta x_n$). In Figure 4.18 we plot the trend of the AUC for the three cases and in Table 4.1 we report the best AUC values. When including the velocity coefficients, significant improvements are provided, while the acceleration coefficients do not seem to be particularly effective. In particular, when using only the original coefficients, a fewer number of Gaussians tends to saturate the results (about 12). When adding the dynamic coefficients, a higher number of Gaussians is required (around 20-30).

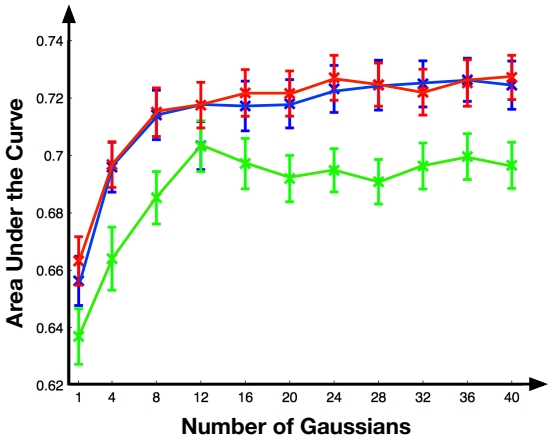


Figure 4.18: Trend of the AUC when varying the number of Gaussian in three cases: (a) using the original coefficient (green line), (b) the original and velocity coefficients (blue line) and (iii) the original, velocity and acceleration coefficients (red line).

	Best AUC
x_n	$70.3 \pm 0.9\%$ $(K = 12)$
$x_n, \Delta x_n$	$72.6 \pm 0.8\%$ $(K = 36)$
$x_n, \Delta x_n, \Delta\Delta x_n$	$72.7 \pm 0.8\%$ $(K = 24)$

Table 4.1: We report the best AUC values with the relative number of Gaussians when comparing the dynamic coefficients.

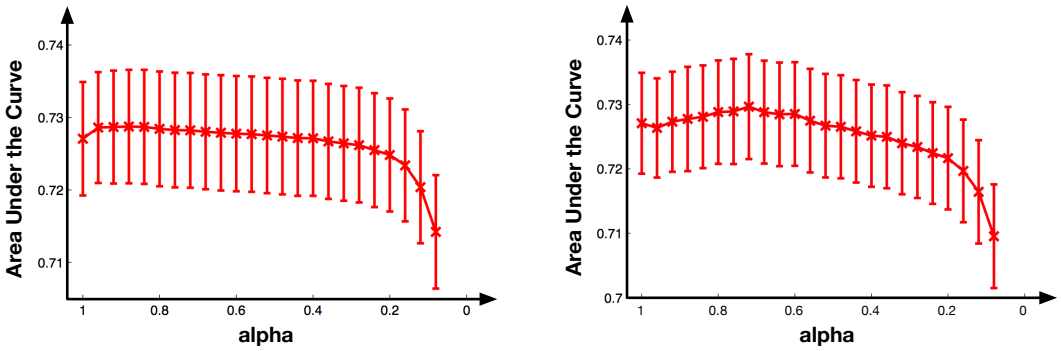


Figure 4.19: The trend of the AUC when varying the alpha parameter of the Universal Background Modeling. On the left part we use all data for training the UBM, on the right part use subpopulation of data for training distinct UBMs and after merge the models together.

4.5.3 Universal Background modeling

The Universal Background Modeling (UBM) is largely used in the speaker recognition context. It is used for representing the speaker independent distribution of the features. The procedure is divided in two step [125]:

- *UBM training.* Many approaches can be used for training an UBM, however two are the most used: (i) to use all data for training the UBM and (ii) to use subpopulation of data for training distinct UBMs and after to merge the models together. While the first method is conceptually easier, it requires a balanced data among the subpopulations. In the other case, the results are biased in the advantage of the more represented subpopulation. The second method on the contrary is robust to the unbalanced subpopulation, however it requires an additional step, the merging of the models.

- *UBM adaptation.* When the UBM model is created, the class models are adapted by using the

UBM. The adaptation is very similar to the EM algorithm. The first step is identical to the expectation step of the EM where the estimates of the sufficient statistics are computed. In the second step, the new estimates are combined with the old estimates by using some adaptation coefficients. In particular, considering a mixture of Gaussians, the j-th mixture is adapted through the following formulas:

$$\hat{w}_j = [\alpha_j^w \frac{n_j}{T} + (1 - \alpha_j^w) w_j]\gamma \tag{4.10}$$

$$\hat{\mu}_j = \alpha_j^m E_j(x) + (1 - \alpha_j^m)\mu_j \tag{4.11}$$

$$\hat{\sigma}_j^2 = \alpha_j^v E_j(x^2) + (1 - \alpha_j^v)(\sigma_j^2 + \mu_j^2) - \hat{\mu}_j^2 \tag{4.12}$$

where $\{\alpha_j^w, \alpha_j^m, \alpha_j^v\}$ are the adaptation coefficients, respectively for the weights, the means and the variances, γ is the scale factor for ensuring that the sum of the weight is equal to 1, n_j is the probabilistic count. As commonly used, we take:

$$\alpha_j^w = \alpha_j^m = \alpha_j^v = \frac{n_j}{n_j + r} \tag{4.13}$$

where r is the *relevance factor*. For mixtures with low probabilistic count, then $\alpha \to 0$ causing the emphasis of the old parameters. In the opposite case, for mixtures with high probabilistic count, then $\alpha \to 1$ causing the emphasis of the new parameters. In practice the relevance factor controls how much new data should be observed by combining the new and old parameters.

We try both training versions, i.e. using all data for training the UBM and dividing the subpopulation of data for training distinct UBMs and after merging the models. We report the results as trend of AUC when varying α_j^w between 1 and 0 with a step of 0.04. In Figure 4.19 we report the trends of the AUC for the first training method (left) and the second training method (right). In the first case, we find that $\alpha_j^w = 0.84$ provides the best results. We obtain an AUC of $72.9 \pm 0.8\%$ AUC. In the second case, we find the best AUC when $\alpha_j^w = 0.72$, yielding an AUC of $73.0 \pm 0.8\%$ AUC. We come up with two conclusions. The first is that for both methods we do not observe significant improvements of AUC, probably due to the fact the UBM is not capable to represent the independent distribution of the features. This can also be due that we use only 24 Gaussians for training the UBM, that is a very low value when compared with other works in the literature. Secondly, the α_i^w value that optimizes the UBM is in both cases relatively close to 1, meaning that the new parameters are emphasized over the old ones.

4.5.4 Evaluation of the features

Generally speaking, there are several reasons for using a feature selection approach. The first one is to reduce the number of features, the overfitting chances and the training time, while improving the generalization of models. The second reason is to better understand the relation between the features and the predicted variables. The feature selection consists in selecting a subset of relevant features and avoiding the redundant or irrelevant ones. Several methods exist for performing the feature selection, however we can distinguish two main classes of algorithms [126]: (i) filter approach and (ii) wrapper approach. In the first case, the selection of the features is independent of the machine learning technique and it is generally considered as pre-processing step. This procedure is generally fast, however sub-optimal subsets can be selected because they are not optimized for the machine learning task. In the other approach, the selection is evaluated through the machine learning technique used for training. As a drawback, this approach could be sensibly slower and the optimal feature subset becomes specific to the machine learning method. In this chapter we use the wrapper approach.

Several types of searching strategies can be used for determining the best set of features. Exponential algorithms, as the exhaustive search, evaluate some subsets that grow exponentially with the size of the search space. Sequential algorithms add or remove one feature sequentially, however they are sensible to local minima. Randomized algorithms include randomness during the searching phase in order to escape local minima. We focus on two sequential algorithms, widely used in literature [127]:

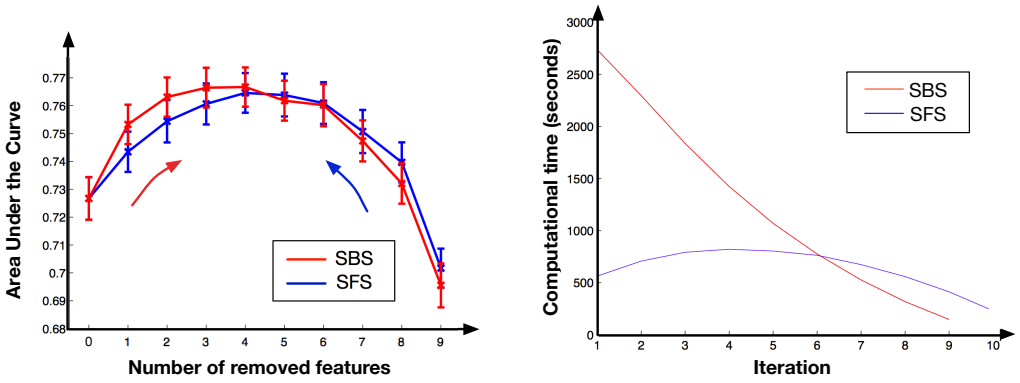


Figure 4.20: Feature selection procedure when using the Sequential Forward Selection (SFS) and the Sequential Backward Selection (SBS) algorithm. In the left part, we report the trend of the average AUC with the confidence interval when reducing the features in the selection set and in the right part, we report the computational time when increasing the iterations.

- *Sequential Forward Selection (SFS)* is the simplest greedy search algorithm. We start with an empty set and add the feature x having the highest objective function $J(Y+x)$, where Y is the set of features already selected and the "+" indicates the feature inclusion in the set. Ideally the objective function J is the probability of correct classification.
- *Sequential Backward Selection (SBS)* is a greedy search algorithm working in the opposite direction of the SFS. We start with the full set of features and we systematically remove the feature x yielding the smallest decrease of the objective function $J(Y-x)$, where Y is the set of features already selected and the "-" indicates the feature removal from the set. As in the previous case, ideally the objective function J is the probability of correct classification.

Different strategies could be used in order to decide the best set of features, for instance some conditions on the number of features or the choice of the objective function. In our case, we choose the set of features yielding to the best AUC value.

We apply the SFS and SBS algorithm using the repeated random sub-sampling validation. In both cases we use the number of random repetitions equal to 100. In particular, the SFS and SBS algorithms are applied on each of the random sub-sampling folder. As a consequence, different sets of features could be chosen and a strategy for determining the best one has to be used. In our specific case, we divide the analysis in two steps:

1. *Feature ranking.* For every repetition we order the features from the most to the less important. As previously said, we could have a different rank depending on the training set under analysis. In order to find a global ranking of features, we assign a score to every feature depending on their rank: given that there are 10 features, we assign 10 points to the first, 9 to the second until 1 point to the last one. We assign the points for every repetition and finally we add the points, obtaining an indication about the "importance" of the features.
2. *Feature selection.* The number of selected features is equal to the number of features that in average yields the best AUC. We finally evaluate the best set of features and the AUC is compared with our baseline result.

In Figure 4.20a, we report the trend of the mean AUC when the number of features in the selection set is reduced for both SBS and SFS. Given that the SFS algorithm adds systematically one feature, we plot the curve in the opposite direction for a better comparison with the SBS algorithm. The arrows indicate the order in which the results are provided for both the algorithms. In both cases, the best AUC is obtained when the number of features is equal to 6. In particular we observe that the performances obtained when several features are present, i.e. less than 6 features, are better for the SBS algorithm, while the opposite occurs when few features are present.

As previously said, we assign a score to every feature depending on their rank, i.e. 10 points to the first, 9 to the second until 1 point to the last one. For every repetition we assign the points and finally we add them, obtaining an indication about the "importance" of the features. In Table 4.2 we report the sequence of features. By considering the first 6 features, we observe that both algorithms provide the same feature list.

	Feature Ranking									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
SFS	8	1	5	6	9	7	10	2	4	3
SBS	8	1	5	9	6	7	10	2	3	4

Table 4.2: Ranking of the features when using the Sequential Forward Selection (SFS) and the Sequential Backward Selection (SBS) algorithm. The numbers correspond to the following features: 1) *amplitude*, 2) *mean*, 3) *minimum*, 4) *maximum*, 5) *standard deviation*, 6) *skewness*, 7) *kurtosis*, 8) *pwmstd*, 9) *ocular pulse amplitude* and 10) *blink density*.

Limiting to 6 features, the same 4 features are also discarded by both algorithms: the *mean*, *minimum*, *maximum* and the *blink density*. All the others are providing useful or not redundant information. Interestingly, the first 3 features appear in the same position for both selection algorithms: the *pwmstd*, *amplitude* and *standard deviation*. We finally test the selected features and we obtain a mean AUC of $74.3 \pm 0.8\%$. However, we have to consider that this result is probably biased by the over-fitting, given that it is obtained on the same data used for selecting the features.

In Figure 4.20b we report the trend of computational time for the SFS (blue) and SBS (red) algorithms when the features are respectively increased or removed. We use the term "iteration" for indicating a step using a specific set of features. Two effects influencing the computational time are present: (i) the feature space dimension and (ii) the number of tested features. In the first case, the training time increases when increasing the dimension of the feature space. For the SFS, for every iteration we increase the feature space, while the opposite occurs for the SBS. In the second case, we have to repeat the training and test for a number of times equals to the number of features nominated for the selection. For both SFS and SBS, the number of features that have to be checked decreases when advancing with the iterations. This explain the behavior of the two curves: the SBS constantly decreases the computational time along with the iterations, while SFS is subjected to two opposing effects, resulting in a maximum at the 4-th iteration.

4.6 Conclusion

In this chapter we present the modeling of physiological data coming from a wearable contact lens sensor (CLS). Our principal aim was to identify the glaucoma illness by using the CLS recordings. The CLS is able to measure 24-hours signals of the variations of the circumferential changes in the area of the corneoscleral junction of the eye. The lens is called Triggerfish® and is produced by Sensimed SA. The measured variations are related to the Intraocular Pressure (IOP), as demonstrated by Sensimed SA. Actually, the only risk factors for the glaucoma largely accepted by the ophthalmologic community are the age, the family history and the IOP level over a specific threshold (21 mmHg). A high IOP is supposed to deform the eyeball in a different manner compared to the standard IOP and the CLS should be able to capture this difference. Sensimed SA gave us access to several CLS recordings of healthy and glaucomatous patients. We tried to discriminate between healthy and primary open angle glaucoma (POAG), which is the most common form of glaucoma.

As first step, we computed three different types of features: statistical, i.e. *amplitude*, *mean*, *minimum*, *maximum*, *standard deviation*, *skewness*, *kurtosis*, physiological, i.e. *ocular pulse amplitude* (OPA), *blink density* and device-based, i.e. *pwmstd*. The feature extraction for obtaining the physiological features was computationally intensive. The OPA is the amplitude of the small variations in CLS recordings due to the hearth rate, i.e. the ocular pulse. Given that blinks introduce a great noise covering the ocular pulse, portions of the CLS recordings not polluted of blinks were extracted. However, the density of blinks in time were computed and used as feature.

In order to compare our machine learning techniques, we used a specific protocol of evaluation, consisting in using a repeated random sub-sampling validation on a rebalanced test set. We randomly selected the 25% of the observations of the less represented class for the test set and an equal number of the other class. The training and test procedures were repeated 100 times. Given that we dealt with an unbalanced set of data in the training and a balanced set of data in the test, we have a different distribution of priors between the two sets. An advantage of generative modeling approach is the natural ability to manage the priors after the training. On the opposite case, discriminative approaches direct model the a posteriori probability, therefore in this case they would be biased in the direction of the most represented class. In the latter case, some re-balancing operations should be undertaken, as the repetition of the observations of the less represented class or filtering some observations of the more represented class. In order to compare the results, we used the Area Under the Curve (AUC) as evaluation metric, given that this metric is particularly used in the medical field.

We applied generative modeling techniques as GMMs and HMMs and we systematically tuned some of its parameters, i.e. the variance floor, the number of states and the number of Gaussians. We found the best set of parameters and we evaluated the influence of the dynamic coefficients on the AUC. By plotting the trend of the AUC when systematically including the dynamic information, we noticed that the velocity coefficients were particularly useful, while the acceleration ones did not seem to add useful information. We performed a feature selection in order to find the best set of parameters. We used two sequential algorithms, i.e. the Sequential Forward Selection and Sequential Backward Selection. The two algorithms respectively added and removed one feature at time. By testing the final set of features, we obtained an AUC of $74.3 \pm 0.8\%$.

We noticed some important facts when using generative modeling approaches in this context. In medical domain, it is useful that the healthcare professionals can interpret the basic machine learning outputs. The meaning of scores emitted by the generative models is simpler than the discriminative one, therefore it could be more easily interpreted by the health-care professionals. The scalability of generative approaches revealed also useful also in this context. As previously said, we discriminated between two classes, i.e. healthy and POAG. However, several other types of glaucoma exist for which an extension of the system is seamless.

Bibliography

[1] V. Rialle, F. Duchene, N. Noury, L. Bajolle, and J. Demongeot, "Health smart home: information technology for patients at home," *Telemedicine Journal and e-health*, vol. 8, no. 4, pp. 395–409, 2002.

[2] World Health Organization, "Definition of health." `http://www.who.int/about/definition/en/print.html`, 2015.

[3] M. Boulos, R. Lou, A. Anastasiou, C. Nugent, J. Alexandersson, G. Zimmermann, U. Cortes, and R. Casas, "Connectivity for healthcare and well-being management: Examples from six european projects," *Int J Environ Res Public Health*, vol. 6, no. 7, pp. 1947–1971, 2009.

[4] H.-H. Chang and J. Moura, "Biomedical signal processin," in *Biomedical Engineerng and Design Handbook*, vol. 1, ch. 22, pp. 559–579, McGraw Hill, 2 ed., 2010.

[5] K. Tripathi, "Important physiological signals in the body." `http://biomedikal.in/2011/05/important-physiological-signals-in-the-body/`, 2011.

[6] J. Bronzino, *The Biomedical Engineering HandBook, Second Edition*. Ed. Joseph D. Bronzino, Boca Raton: CRC Press LLC, 2010.

[7] A. Pantelopoulos and N. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, pp. 1–12, Jan 2010.

[8] O. Lara and M. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[9] E. Tapia, S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, pp. 37–40, Oct 2007.

[10] J. Pärkkä, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 119–128, Jan 2006.

[11] O. Lara, A. Pérez, M. Labrador, and J. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive and Mobile Computing*, vol. 8, no. 5, pp. 717–729, 2012.

[12] A. Bulling, J. Ward, H. Gellersen, and G. Troster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 741–753, April 2011.

[13] S. Jerritta, M. Murugappan, R. Nagarajan, and K. Wan, "Physiological signals based human emotion recognition: a review," in *Proceedings of the IEEE 7th International Colloquium on Signal Processing and its Applications (CSPA)*, pp. 410–415, March 2011.

[14] K. Kim, S. Bang, and S. Kim, "Emotion recognition system using short-term monitoring of physiological signals," *Medical and Biological Engineering and Computing*, vol. 42, no. 3, pp. 419–427, 2004.

[15] J. Wagner, K. Jonghwa, and E. Andre, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 940–943, July 2005.

[16] C. Liu, P. Rani, and N. Sarkar, "An empirical study of machine learning techniques for affect recognition in human-robot interaction," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2662–2667, Aug 2005.

[17] O. Barnea and V. Shusterman, "Analysis of skin-temperature variability compared to variability of blood pressure and heart rate," in *Proceedings of the 17th IEEE Annual Conference on Engineering in Medicine and Biology Society*, vol. 2, pp. 1027–1028, Sep 1995.

[18] H. Kataoka, H. Kano, H. Yoshida, A. Saijo, M. Yasuda, and M. Osumi, "Development of a skin temperature measuring system for non-contact stress evaluation," in *Engineering in Medicine and Biology Society, Proceedings of the 20th Annual International Conference of the IEEE*, vol. 2, pp. 940–943, Oct 1998.

[19] K. Jonghwa and E. Andre, "Emotion recognition based on physiological changes in music listening," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 2067–2083, Dec 2008.

[20] I. Homma and Y. Masaoka, "Breathing rhythms and emotions," *Experimental Physiology*, vol. 93, no. 9, pp. 1011–1021, 2008.

[21] J. Cacioppo, G. Berntson, and J. Larsen, *The psychophysiology of emotion.* R. Lewis, J.M. Haviland-Jones (Eds.), 2000.

[22] B. Wexler, S. Warrenburg, G. Schwartz, and L. Janer, "EEG and EMG responses to emotion-evoking stimuli processed without conscious awareness," *Neuropsychologia*, vol. 30, no. 12, pp. 1065–1079, 1992.

[23] L. Biel, O. Pettersson, L. Philipson, and P. Wide, "Ecg analysis: a new approach in human identification," *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 3, pp. 808–812, 2001.

[24] T. Shen, W. Tompkins, and Y. Hu, "One-lead ecg for identity verification," in *Proceedings of the 2nd Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference in Engineering in Medicine and Biology*, vol. 1, pp. 62–63, 2002.

[25] W. Wang, H. Wang, M. Hempel, D. Peng, H. Sharif, and H.-H. Chen, "Secure stochastic ecg signals based on gaussian mixture model for e-healthcare systems," *Systems Journal, IEEE*, vol. 5, no. 4, pp. 564–573, 2011.

[26] Y. Gu, Y. Zhang, and Y. Zhang, "A novel biometric approach in human verification by photoplethysmographic signals," in *Proceedings of the 4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, pp. 13–14, April 2003.

[27] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS)*, pp. 2455–2458, 2005.

[28] J. Cannan and H. Huosheng, "Automatic user identification by using forearm biometrics," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 710–715, July 2013.

[29] M. Suresh, P. Krishnamohan, and M. Holi, "Gmm modeling of person information from emg signals," in *Proceedings of the Conference on Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 712–717, Sept 2011.

[30] B.-S. Lin, N.-K. Chou, F.-C. Chong, and S.-J. Chen, "Rtwpms: A real-time wireless physiological monitoring system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 647–656, Oct 2006.

[31] C. Linti, H. Horter, P. Osterreicher, and H. Planck, "Sensory baby vest for the monitoring of infants," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 134–137, April 2006.

[32] E.-M. Strehle, W. Gray, S. Gopisetti, J. Richardson, J. McGuire, and S. Malone, "Can home monitoring reduce mortality in infants at increased risk of sudden infant death syndrome? a systematic review," *Nature Communication*, vol. 101, 2012.

[33] J. Habetha, "The myheart project–fighting cardiovascular diseases by prevention and early diagnosis," in *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEMBS)*, pp. 6746–6749, 2006.

[34] M. Baig, H. Gholamhosseini, and M. Connolly, "A comprehensive survey of wearable and wireless ecg monitoring systems for older adults," *Medical & Biological Engineering & Computing*, vol. 51, no. 5, pp. 485–495, 2013.

[35] R. Fensli, E. Gunnarson, and T. Gundersen, "A wearable ecg-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation," in *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, pp. 407–412, June 2005.

[36] T.-B. Nguyen, W. Lou, T. Caelli, S. Venkatesh, and D. Phung, "Individualized arrhythmia detection with ecg signals from wearable devices," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 570–576, Oct 2014.

[37] J. Luprano, J. Sola, S. Dasen, J.-M. Koller, and O. Chetelat, "Combination of body sensor networks and on-body signal processing algorithms: the practical case of myheart project," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 4 pp.–79, April 2006.

[38] Y.-D. Lee and W.-Y. Chung, "Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring," *Sensors and Actuators B: Chemical*, vol. 140, no. 2, pp. 390–395, 2009.

[39] U. Anliker, J. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic, "Amon: a wearable multiparameter medical monitoring and alert system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 415–427, Dec 2004.

[40] P. Maulik and C. Harbour, "Epidemiology of intellectual disability," *International Encyclopedia of Rehabilitation, JH Stone, M Blouin, editors*, 2010.

[41] J. Langlois, W. Rutland-Brown, and M. Wald, "The epidemiology and impact of traumatic brain injury: a brief overview.," *the Journal of Head Trauma rehabilitation*, vol. 21, no. 5, pp. 375–378, 2006.

[42] A. Tura, M. Badanai, D. Longo, and L. Quareni, "A medical wearable device with wireless bluetooth-based data transmission," *Measurement Science Review*, vol. 3, pp. 1–4, 2003.

[43] A. Tura, M. Badanai, D. Longo, and L. Quareni, "A multi-functional, portable device with wireless transmission for home monitoring of children with a learning disability," *Journal of Telemedicine and Telecare*, vol. 10, no. 5, pp. 298–302, 2004.

[44] G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, R. Paradiso, and E. Scilingo, "Wearable monitoring for mood recognition in bipolar disorder based on history-dependent long-term heart rate variability analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1625–1635, Sept 2014.

[45] D. Mahoney, R. Purtilo, F. Webbe, M. Alwan, A. Bharucha, T. Adlam, H. Jimison, B. Turner, and S. Becker, "In-home monitoring of persons with dementia: Ethical guidelines for technology research and development," *Alzheimer & Dementia*, vol. 3, no. 3, pp. 217–226, 2007.

[46] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Wearable eye tracking for mental health monitoring," *Computer Communications*, vol. 35, no. 11, pp. 1306 – 1311, 2012.

[47] I. Bruderman and S. Abboud, "A portable forced oscillation device for respiratory home monitoring," *Telemedicine Journal*, vol. 3, no. 2, pp. 127–133, 1997.

[48] J. Rigau, R. Farre, J. Roca, S. Marco, A. Herms, and D. Navajas, "A portable forced oscillation device for respiratory home monitoring.," *The European Respiratory Journal*, vol. 19, no. 1, pp. 146–150, 2002.

[49] B. Dinesen, L. Haesum, N. Soerensen, C. Nielsen, O. Grann, O. Hejlesen, E. Toft, and L. Ehlers, "A portable forced oscillation device for respiratory home monitoring," *Journal of Telemedicine and Telecare*, vol. 18, no. 4, pp. 221–225, 2012.

[50] J. Junior, M. Rabahi, M. Conde, K. Corra, H. Rabahi, and A. Rocha, "Pao2 as a predictive factor for sleep respiratory disorders in patients with chronic obstructive pulmonary disease," *Chest Journal*, vol. 148, no. 4, 2015.

[51] Y. Nakayama-Ashida, M. Takegami, K. Chin, K. Sumi, T. Nakamura, K.-I. Takahashi, T. Wakamura, S. Horita, Y. Oka, I. Minami, S. Fukuhara, and H. Kadotani, "Sleep-disordered breathing in the usual lifestyle setting as detected with home monitoring in a population of working men in japan," *Sleep*, vol. 31, no. 3, pp. 419–425, 2008.

[52] H. Le, T. Dang, and F. Liu, "Towards long-term large-scale visual health monitoring using cyber glasses," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth, pp. 200–207, 2013.

[53] D. Robinson, "A method of measuring eye movemnent using a scieral search coil in a magnetic field," *IEEE Transactions on Bio-medical Electronics*, vol. 10, pp. 137–145, Oct 1963.

[54] K. Kim, S. Lee, and H. Kim, "A wireless measurement system for three-dimensional ocular movement using the magnetic contact lens sensing technique," in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEMBS)*, vol. 1, pp. 2287–2289, Sept 2004.

[55] R. Badugu, J. Lakowicz, and C. Geddes, "Noninvasive continuous monitoring of physiological glucose using a monosaccharide-sensing contact lens," *Analytical chemistry*, vol. 76, no. 3, pp. 610–618, 2004.

[56] W. March, A. Mueller, and P. Herbrechtsmeier, "Clinical trial of a noninvasive contact lens glucose sensor," *Diabetes Technology & Therapeutics*, vol. 6, no. 6, pp. 782–789, 2004.

[57] V. Alexeev, S. Das, D. Finegold, and S. Asher, "Photonic crystal glucose-sensing material for noninvasive monitoring of glucose in tear fluid.," *Chimical Chemistry*, vol. 50, no. 12, pp. 2353–60, 2004.

[58] M. Chu, K.M., D. Takahashi, T. Arakawa, K. Sano, S.-I. Sawada, H. Kudo, Y. Iwasaki, K. Akiyoshi, M. Mochizuki, and K. Mitsubayashi, "Soft contact lens biosensor for in situ monitoring of tear glucose as non-invasive blood sugar assessment," *Talanta*, vol. 83, no. 3, pp. 960–965, 2011.

[59] H. Yao, C. Marcheselli, A. Afanasiev, I. Lahdesmaki, and B. Parviz, "A soft hydrogel contact lens with an encapsulated sensor for tear glucose monitoring," in *Proceedings of the 25th IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pp. 769–772, Jan 2012.

[60] H. Yao, A. Afanasiev, I. Lahdesmaki, and B. Parviz, "A dual microscale glucose sensor on a contact lens, tested in conditions mimicking the eye," in *Proceedings of the 24th IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pp. 25–28, Jan 2011.

[61] M. Senior, "Novartis signs up for google smart lens," *Nature Biotechnology*, vol. 32, 2014.

[62] R. Mapstone, "Determinants of corneal temperature," *The British Journal of Ophthalmology*, vol. 52, no. 10, pp. 729–741, 1968.

[63] J. Kinn and R. Tell, "A liquid-crystal contact-lens device for measurement of corneal temperature," *Biomedical Engineering, IEEE Transactions on*, vol. BME-20, pp. 387–388, Sept 1973.

[64] V. Scott, L. Tarassenko, C. Glynn, and A. Hill, "Contact lens oximetry: a valid concept?," in *Pulse Oximetry: A Critical Appraisal, IEE Colloquium on*, pp. 1–3, May 1996.

[65] J. Roodhooft, "Leading causes of blindness worldwide," *Bulletin de la Societe Belge d'Ophtalmologie*, vol. 283, pp. 19–25, 2002.

[66] N. Congdon, "Prevention strategies for age related cataract: present limitations and future possibilities," *The British Journal of Ophthalmology*, vol. 85, no. 5, pp. 516–520, 2001.

[67] H. Quigley and A. Broman, "The number of people with glaucoma worldwide in 2010 and 2020," *The British Journal of Ophthalmology*, vol. 90, no. 3, pp. 262–267, 2006.

[68] A. Van Gestel, C. Webers, H. Beckers, M. Van Dongen, J. Severens, F. Hendrikse, and J. Schouten, "The relationship between visual field loss in glaucoma and health-related quality-of-life," *Royal College of Ophthalmologists*, vol. 24, no. 12, pp. 262–267, 2010.

[69] J. Tielsch, J. Katz, K. Singh, H. Quigley, J. Gottsch, J. Javitt, and A. Sommer, "A population-based evaluation of glaucoma screening: the baltimore eye survey," *American Journal of Epidemiology*, vol. 134, no. 10, 1991.

[70] R. Weinreb and P. Khaw, "Primary open-angle glaucoma," *Lancet*, vol. 363, pp. 1711–1720, 2004.

[71] M. Shields, "Normal-tension glaucoma: is it different from primary open-angle glaucoma?," *Current Opinion in Ophthalmology*, vol. 19, pp. 85–88, 2008.

[72] S. Mroczkowska, A. Benavente-Perez, A. Negi, V. Sung, S. Patel, and D. Gherghel, "Primary Open-Angle Glaucoma vs Normal-Tension Glaucoma," *Jama Ophthalmology*, vol. 131, no. 1, pp. 36–43, 2013.

[73] R. David, L. Zangwill, D. Briscoe, M. Dagan, R. Yagev, and Y. Yassur, "Diurnal intraocular pressure variation: an analysis of 690 curves," *The British Journal of Ophthalmology*, vol. 76, pp. 280–283, 1992.

[74] C. De Vivero, C. O'Brien, L. Lanigan, and R. Hitchings, "Diurnal intraocular pressure variation in low-tension glaucoma," *Eye*, vol. 8, pp. 521–523, 1994.

[75] E. Hughes, P. Spry, and J. Diamond, "24-hour monitoring of intraocular pressure in glaucoma management: a retrospective review," *Journal of Glaucoma*, vol. 12, no. 3, pp. 232–236, 2003.

[76] K. Katuri, S. Asrani, and M. Ramasubramanian, "Intraocular pressure monitoring sensors," *Sensors Journal, IEEE*, vol. 8, pp. 12–19, Jan 2008.

[77] C. Collins, "Miniature passive pressure transensor for implanting in the eye," *IEEE Transactions on Biomedical Engineering*, vol. BME-14, no. 2, pp. 74–83, 1967.

[78] "Sensimed AG, Lausanne, Switzerland." `http://www.sensimed.ch`, 2015.

[79] "Sensimed triggerfish." http://www.sensimed.ch/en/sensimed-triggerfish/sensimed-triggerfish.html, 2015.

[80] "Ziemer Ophtalmology." http://www.ziemergroup.com/, 2015.

[81] Ziemer Ophtalmology, "Dct-lens." http://www.ziemergroup.com/products/pascal/dct-lens.html, 2015.

[82] M. Twa, C. Roberts, H. Karol, A. Mahmoud, P. Weber, and R. Small, "Evaluation of a contact lens-embedded sensor for intraocular pressure measurement," *Journal of Glaucoma*, vol. 19, no. 6, pp. 382–390, 2010.

[83] G. Salvatore, N. Mnzenrieder, T. Kinkeldei, L. Petti, C. Zysset, I. Strebel, L. Bthe, and G. Trster, "Wafer-scale design of lightweight and transparent electronics that wraps around hairs," *Nature Communication*, vol. 5, 2013.

[84] J. Yan, "An unpowered, wireless contact lens pressure sensor for point-of-care glaucoma diagnosis," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2522–2525, Aug 2011.

[85] G.-Z. Chen, I.-S. Chan, and D. Lam, "Capacitive contact lens sensor for continuous non-invasive intraocular pressure monitoring," *Sensors and Actuators A: Physical*, vol. 203, pp. 112 – 118, 2013.

[86] G.-Z. Chen, I.-S. Chan, L. Leung, and D. Lam, "Soft wearable contact lens sensor for continuous intraocular pressure monitoring," *Medical Engineering & Physics*, vol. 36, no. 9, pp. 1134–1139, 2014.

[87] H. Cong and T. Pan, "Microfabrication of conductive pdms on flexible substrates for biomedical applications," in *Proceedings of the 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems (NEMS)*, pp. 731–734, Jan 2009.

[88] K. Mansouri and T. Shaarawy, "Continuous intraocular pressure monitoring with a wireless ocular telemetry sensor: initial clinical experience in patients with open angle glaucoma," *The British Journal of Ophthalmology*, vol. 95, no. 5, pp. 627–629, 2011.

[89] J. Roberts, "Circadian rhythm and human health." http://www.photobiology.info/Roberts-CR.html, 2010.

[90] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *IEEE Transaction on Biomedical Engineering*, vol. 37, no. 9, pp. 826–36, 1990.

[91] R. Andreao, B. Dorizzi, and J. Boudy, "Ecg signal analysis through hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 1541–1549, Aug 2006.

[92] A. Koski, "Modelling {ECG} signals with hidden markov models," *Artificial Intelligence in Medicine*, vol. 8, no. 5, pp. 453 – 471, 1996.

[93] M. Altuve, G. Carrault, A. Beuchee, P. Pladys, and A. Hernasndez, "Online apneabradycardia detection based on hidden semi-markov models," *Medical & Biological Engineering & Computing*, vol. 53, no. 1, pp. 1–13, 2015.

[94] A. Chan, K. Englehart, B. Hudgins, and D. Lovely, "Hidden markov model classification of myoelectric signals in speech," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, pp. 143–146, Sept 2002.

[95] H. Manabe and Z. Zhang, "Multi-stream hmm for emg-based speech recognition," in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEMBS)*, vol. 2, pp. 4389–4392, Sept 2004.

[96] M. Meng, Q. She, Y. Gao, and Z. Luo, "Emg signals based gait phases recognition using hidden markov models," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA)*, pp. 852–856, June 2010.

[97] M. Caon, F. Carrino, A. Ridi, Y. Yue, O. Khaled, and E. Mugellini, "Kinesiologic electromyography for activity recognition," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA, pp. 34:1–34:7, ACM, 2013.

[98] J. Chiang, J. Wang, and M. McKeown, "A hidden markov, multivariate autoregressive (hmm-mar) network framework for analysis of surface emg (semg) data," *IEEE Transactions on Signal Processing*, vol. 56, pp. 4069–4081, Aug 2008.

[99] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, 2007.

[100] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller, "Hidden Markov models for online classification of single trial EEG data," *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1299 – 1309, 2001.

[101] S. Zhong and J. Ghosh, "Hmms and coupled hmms for multi-channel eeg classification," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, vol. 2, pp. 1154–1159, 2002.

[102] S. Chiappa and S. Bengio, "Hmm and iohmm modeling of eeg rhythms for asynchronous bci systems," in *European Symposium on Artificial Neural Networks ESANN*, 2004.

[103] D. Lederman and J. Tabrikian, "Classification of multichannel eeg patterns using parallel hidden markov models," *Medical & Biological Engineering & Computing*, vol. 50, no. 4, pp. 319–328, 2012.

[104] T. Al-Ani, Y. Hamam, R. Fodil, F. Lofaso, and D. Isabey, "Using hidden markov models for sleep disordered breathing identification," *Simulation Modelling Practice and Theory*, vol. 12, no. 2, pp. 117 – 128, 2004.

[105] H. Ravishankar, A. Saha, G. Swamy, and S. Genc, "An early respiratory distress detection method with markov models," in *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3438–3441, Aug 2014.

[106] M. Leonardi, P. Leuenberger, D. Bertrand, A. Bertsch, and P. Renaud, "First steps toward noninvasive intraocular pressure monitoring with a sensing contact lens," *Investigative Ophthalmology & Visual Science (IOVS)*, vol. 45, no. 9, pp. 3113–3117, 2004.

[107] M. Galin, I. Baras, and M. Best, "The nature of the ocular pulse," *The Journal of Head and Face Pain*, vol. 9, pp. 112–128, 1969.

[108] S. Hayreh, "Orbital vascular anatomy," *Eye*, vol. 20, no. 10, pp. 1130–1144, 2006.

[109] K. Schmidt, A. Von Rckmann, and T. Mittag, "Ocular pulse amplitude in ocular hypertension and open-angle glaucoma," *International Journal of Ophthalmology*, vol. 212, pp. 5–10, 1998.

[110] O. Schwenn, R. Troost, A. Vogel, F. Grus, S. Beck, and N. Pfeiffer, "Ocular pulse amplitude in patients with open angle glaucoma, normal tension glaucoma, and ocular hypertension," *British Journal of Ophthalmology*, vol. 86, pp. 981–985, 2002.

[111] O. Punjabi, H.-K. Ho, and C. Kniestedta, "Intraocular pressure and ocular pulse amplitude comparisons in different types of glaucoma using dynamic contour tonometry," *Current Eye Research*, vol. 31, pp. 851–862, 2006.

[112] I. Stalmans, A. Harris, V. Vanbellinghen, T. Zeyen, and B. Siesky, "Ocular pulse amplitude in normal tension and primary open angle glaucoma," *Journal of Glaucoma*, vol. 17, pp. 403–407, 2008.

[113] National Instruments, "Zero padding does not buy spectral resolution." `http://www.ni.com/tutorial/4880/en/`, 2006.

[114] A. Kamshilin, S. Miridonov, V. Teplov, R. Saarenheimo, and E. Nippolainen, "Photoplethysmographic imaging of high spatial resolution," *Biomed Opt Express*, vol. 2, pp. 996–1006, 2011.

[115] M. Johnstone, E. Martin, and A. Jamil, "Pulsatile flow into the aqueous veins: manifestations in normal and glaucomatous eyes," *Experimental Eye Research*, vol. 92, no. 5, pp. 318–327, 2011.

[116] L. An, J. Chao, M. Johnstone, and R. Wang, "Noninvasive imaging of pulsatile movements of the optic nerve head in normal human subjects using phase-sensitive spectral domain optical coherence tomography," *Optics Letters*, vol. 38, pp. 1512–1514, 2013.

[117] C. Gisler, A. Ridi, J. Hennebert, R. Weinreb, and K. Mansouri, "Automated detection and quantification of circadian eye blinks using a contact lens sensor," *Translation Vision Science and Technology*, vol. 4, no. 1, pp. 1512–1514, 2015.

[118] F. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

[119] R. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, pp. 55–63, 2002.

[120] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764 – 766, 2013.

[121] K. Chrominski and M. Tkacz, "Comparison of outlier detection methods in medical data," *Journal of Medical Informatics & Technologies*, vol. 16, pp. 89 – 94, 2010.

[122] J. Hanley and B. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[123] C. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, pp. 283–298, 1978.

[124] H. Melin and J. Lindberg, "Variance flooring, scaling and tying for text-dependent speaker verification," *Proceedings of Eurospeech*, 1999.

[125] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 13, pp. 19 – 41, 2000.

[126] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 12, pp. 273–324, 1997.

[127] R. Gutierrez-Osuna, "Lecture 11: Sequential Feature Selection." `http://www.ni.com/tutorial/4880/en/`.

Chapter 5

Activity Recognition for Energy Savings

Contents

5.1 Introduction

In developed countries, the buildings energy demand represents one of the major source of consumption. In particular, in the EU and USA, the energy consumption of buildings is between 20% and 40% of the total energy consumption. In buildings, the system for heating, ventilation and air conditioning (HVAC) represents the major source of consumption, reaching about half of the total. The electric lighting and appliances are also important sources of consumption, especially in offices, where they reach together about 35% of the total [1]. In the near future an increment of the energy demand is

expected, mainly due to growth in population and an increasing demand for building services and comfort levels.

Generally speaking, smart houses and automated buildings aim to increase human comfort while reducing energy consumption. Services are provided as security of the inhabitants through devices with automatic functions and/or remotely controlled systems [2]. Other applications in the medical domain have been presented [3, 4]. In particular, health monitoring and home-automated assistance are more and more relevant due to population aging and multiple kinds of disabilities. The monitoring can be helpful to detect uncommon situations and the automated assistance helps the inhabitants to perform tasks otherwise hard to make [2]. From the energetic and economic point of view, the status of the HVAC system, lighting and appliances can be related to specific energy-demanding activities, as cooking or housekeeping. Knowledge about activities provides valuable information about energy consumption. If human activities are known, energy demand can be modeled for predicting and optimizing the energy consumption [5].

Improving human comfort usually affects the energy consumption, for instance by using the heating system for warming up the house when occupants are present. However, in realistic scenarios, energy wasting may occur, for instance if warming up the environment when occupants are absent. Information about activity recognition and prediction in buildings can be useful for optimizing the energy consumption and comfort. For example, lowering the set-point temperature of the heating control system when occupants are absent, sleeping or cooking can lead to important energy savings while at the same time the thermal comfort is maintained.

In this chapter we aim at the elaboration of Information Systems able to optimize energy consumption in buildings while preserving human comfort. We use stochastic state-based modeling on temporal signals acquired from heterogeneous sources, as distributed sensors. These temporal signals have a state-based nature, corresponding to "modes" of use of the building, house or equipment. These "modes" depend on the activity performed by users inside their habitations. We model these signals with state-based stochastic approaches such as GMMs and HMMs. Such models allow detecting automatically probable states related to modes of use that will then allow a better parameterization of building control algorithms. More specifically, we report on the design, conception and validation of machine learning models and we show the pertinence of the state-based modeling in buildings when using HMMs. In this chapter, we will improve the modeling performances by relaxing the Markovian hypothesis through the inclusion of the information on the activity duration. Future activities are predicted by using the information coming from the state-based modeling. This system allows to reduce the energy consumption while at the same time preserving the user comfort.

5.2 State of the Art

5.2.1 Smart Home

Definition

For sake of simplicity, we use the term "Smart Home" for indicating an environment where an automatic system is able to take "Smart" decisions. Several definitions are present in literature, systematically delineating the field of application and specifying the "smartness" of the system.

Alam et al. [6] define the Smart Home as "*an application of ubiquitous computing that is able to provide user context-aware automated or assistive services in the form of ambient intelligence, remote home control, or home automation*". They divide the services provided to the inhabitants in three categories: comfort, healthcare and security. While the first two can be provided remotely, security measures are locally provided, allowing access to authorized users. Yiu et al. [7] list some important characteristics of a Smart Home: comfort, health, safety and security of the users, the sustainability and cost-effectiveness of achieving the performance requirements. Considering these goals, the difference between an intelligent and an unintelligent home is the capability of self-adapting to the dynamic environmental changes. In addition, they make a parallel between a Smart Home and the human intelligence: some characteristics, i.e. control, memory, learning, feedback and monitoring, automatic response and choice of alternative responses are shared between both systems. Aldrich [8] distinguishes five types of Smart Homes depending on the level of "intelligence": (i) *homes which*

contain intelligent objects, (ii) *homes which contain intelligent, communicating objects*, (iii) *connected homes*, (iv) *learning homes*, (v) *attentive homes*.

In this chapter, we mainly focus on the application of energy saving measures while preserving human comfort. These two aspects are tightly related: for instance, focusing only on the energy saving aspect without considering the human comfort would have the consequence of switching off all the sources of consumption, probably making the home unfit for residents. The compromise between energy optimization while preserving human comfort can be shortly indicated by the term "energy efficiency" .

Network

The *Intelligent Building Management System* (IBMS), also called *Building Automation System* (BAS), is constituted by the hardware and software infrastructures dedicated to the control and optimization of the building mechanical and electrical equipment. It describes the advanced functionality provided by the control system of a building. The aim of an IBMS is to reduce building energy and costs compared to a non-controlled building. Usually this aim is reached through the control of the lighting, heating, air conditioning, water, alarm (and security) and occupancy revelation systems.

An IBMS is constituted by several hardware-based elements:

- **Sensors** (or detectors) are converters that measure a physical quantity and convert it into a signal which can be read by a given instrument. Three types of metering systems can be installed in a Smart Home: (i) wearable devices, (ii) direct environment sensors and (iii) infrastructure mediated sensors [9]. In the first approach, sensors as wearable accelerometers and RFID are worn by residents. Precise information about the resident positions and interactions can be retrieved. However, as a drawback, the sensors have to be always worn. In the second approach, a set of sensors is placed in the environment. In the third approach, the existing home infrastructure, e.g. the electrical system, is used to mediate the events. In this chapter, we will focus on the second approach.

  The direct environment sensors can be separated in three categories: (i) simple binary sensors, (ii) metering sensors and (iii) recording devices. Simple binary sensors give a binary indication, "0" or "1", when a certain condition is verified. Motion detectors are a typical example of this kind of technology: when a movement is detected, the state change from "0" to "1". After a certain time, if no more movements are detected, the state change from "1" to "0". Metering sensors are used for measuring a physical variable changing in time. In some cases the measure is made at a fixed interval of time, in other cases an event triggers the measurement. Temperature sensors are an example of this kind of technology. In the third category devices as cameras and microphones provide high-level content data. In particular, the vision-based approaches mostly use computer vision techniques based on video stream analysis from regular, stereo or even infrared cameras. Tracking the movements or positions of the users in the house is often a target application of vision-based systems. Their drawback is in a rather heavy computational load and sensitivity to illumination variations. Moreover problems of privacy are addressed and these devices are not easily accepted by residents in real-context scenarios. In this chapter we will focus on data collected by the first two categories, that, for sake of simplicity, we will call *simple environment sensors*.

  Given the democratization of the IoT technology, in our opinion Smart Homes will more and more use a multitude of small, low-cost and easy-to-install sensors. In fact, the most common sensors used in Smart Home are: presence / motion, temperature, light, electricity, windows and doors, humidity, gas sensor.

- **Actuators** are devices used for actuating the decisions taken by the IBMS. In some cases the actuators are activated depending on internal environmental factors (temperature, humidity, illumination, chemicals in air) or users' physiological comfort and psychological responses (pleasure, anger, sadness, happiness). In other cases actuators can be absent, for instance when the aim of the Smart Home is only to provide a feedback to the users. Among the most common actuators we find automatic thermostats, light controllers, store controllers, power switches and appliance controller.

- **Decision center** is the unit responsible of the decision making. After having collected data provided by sensors, it applies the decisions through the actuators. It is usually composed of two parts: (i) a central memory unit, where the information is physically stored and (ii) a processing unit, for data analysis and decision making.
- **Network protocol**. Hardware-based elements are communicating the one with the others. Among the most common standards we find the European Installation Bus (EIB), KNX and IP [10].

One of the first Smart Home realized is the Neural Network House by Mozer [11]. He implements the Adaptive Control of Home Environment with the aim of anticipating the inhabitants' needs while managing the energy conservation. In particular the system controls the HVAC system, lighting and water heating. The sensors permit to have information about the status of lights, fans, temperature, ambient illumination, room temperature, sound level, motion detector activity, status of the doors and windows (if open or closed) and other information such as water heater, outdoor temperature and weather. The actuators have the ability to control status and intensity of light banks, status and speed of ceiling fans, status of water heater, status of gas furnace, status of electric space heaters and status of speakers in each room through which computer can communicate. Few years later, Hagras et al. [12] realize a system where each room contains an embedded-agent. Every embedded agent is connected through the IP network, while inside every room the sensors and actuators are connected to the embedded-agent. This network permits the adaptation to the building configuration and a room-based distribution of the computation.

5.2.2 Energy saving potential

As a matter of fact, in many buildings (residential or not) a considerable amount of energy can potentially be saved. This amount of energy depends on several factors, as the smartness of the decisions, the desired comfort and the type of building. The first aspect will be further investigated, while the latter will be analyzed in the Appendix B.

In order to understand the potential energetic and economical gain of a Smart Home, we firstly should be able to answer to the question: how much energy can we potentially save ? For being a favorable investment, the cost of an IBMS, in terms of energy consumption and money, has to be payed back by the saved energy.

As stated in the introduction, the global contribution from buildings, both residential and commercial, toward energy consumption in developed countries has reached a rate varying between 20% and 40% of the total [1]. These values are growing because of the growth of the population, the increasing demand for comfort and building services, and the rise of the time spent inside buildings. By investigating the different sub-systems of a Smart Home, more detailed information can be provided:

- The *HVAC systems* energy use is very intensive: in USA, it represents 50% of the building consumption and 20% of the total consumption. The HVAC system is one of the major source of consumption given the huge amount of energy needed. For this reason, the energy saving when dealing with HVAC systems in some cases can attain 60% of the total energy consumption [13].
- The *artificial lighting* is estimated to account for 25-40% of the energy consumption of commercial buildings in the USA [14]. Other authors [1] estimate the lighting weight as 15% in non-domestic buildings. Using some simple occupancy sensors for controlling lighting, the energy saved was up to 30% on the total energy used for lighting [15].
- The *appliances* are estimated as consuming about 10% of the total energy in non-domestic buildings [1].

The energy consumption not only depends on how the energy is used, i.e. the optimization of the available resources and the desired comfort, but also on the physical characteristics of the building. In particular the insulation and the daylight exposition play an important role and some Energy Conservation Measures can be adopted on the building. For instance the insulation of external walls contributes for 33-60% energy savings, weather proofing of openings for 16-21%, the installation of double-glazed windows for 14-20%, and the regular maintenance of central heating boilers contributes for 10-12% [16].

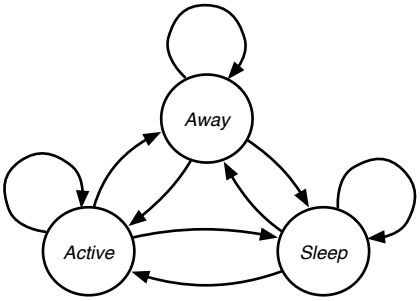


Figure 5.1: Example of a simple presence recognition model using three states: *Active*, *Away* and *Sleep*.

In average a significant amount of energy can potentially be saved by adopting an IBMS solutions or directly intervening on the structure of the building. Unfortunately these solutions are not as popular as they should be. This can be explained by the fact that often people do not want to afford the installation costs that will be re-payed after an undetermined and potentially large number of years. On the other hand, from the environmental perspective, if these solutions would be adopted by a large part of the population, every year billion tons of pollutants for generating the surplus energy would not be released in our environment.

5.2.3 State based-modeling in Smart Home

In order to build a system able to take smart decisions, data representing the process under analysis needs first to be collected. In a second step, data can be used for searching and recognizing recurrent patterns, typically through the application of machine learning techniques. For instance patterns related to human activities could be used for understanding the human behavior and applying energy saving measures.

In this chapter we focus on the application of state-based modeling on data collected by using simple environmental sensors. Stochastic models, as Bayesian networks and HMMs, are often used for recognition purposes in Smart Home context [17]. We report on different recognition tasks performed: (i) presence recognition, (ii) activity recognition, (iii) resident recognition and (iv) identification of the occupied spaces.

Presence recognition

The aim of presence recognition is to detect the presence of inhabitants inside the Smart Home. Very simple sensors as motion detectors can be used for this task. If the motion sensor detects a movement in the area under analysis, it switches to the *Active* state. It waits for a silent period and after it switch to the *Idle* state. The *Active* and *Idle* states can be related to simple user activities: for instance, depending on the daytime, *Idle* can be related to *Away* during the day, and *Sleep* during the night. Night and day time can be a priori defined, for instance it can be considered night from midnight to 8 AM. A simple controller using this information could already provide important energy savings. For instance, it has been shown that 20-30% of energy can be saved by turning off the HVAC system when the residents are sleeping or away [18].

An example of this approach has been shown by Lu et al. [19]. They create a smart thermostat, which uses information about the users' presence in order to turn off the HVAC system. For their analysis they use motion and door sensors and a 3-states HMM, represented in Figure 5.1. The three states are: *Away*, when no one is inside the home, *Active*, if at least one person is inside the home and *Sleep* when all residents are sleeping. Every five minutes the HMM transits to a new state. They use as features the time of a day with a 4 hours window of data acquired by the motion and door sensors. The HMM can correctly classify more than 90% of the instances and they achieve an energy saving of about 28%.

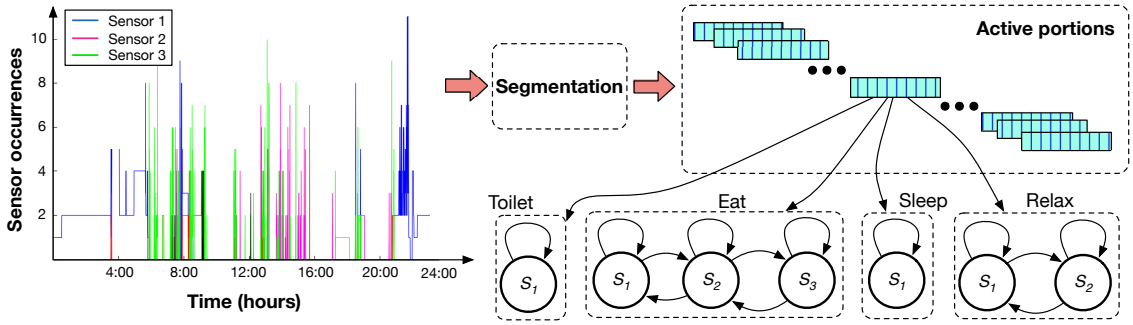


Figure 5.2: Activity recognition with a procedure of windowing with a priori segmentation. The segmentation step detects portions of the "active" signal. Each window is then tested against several GMMs or HMMs representing a given activity.

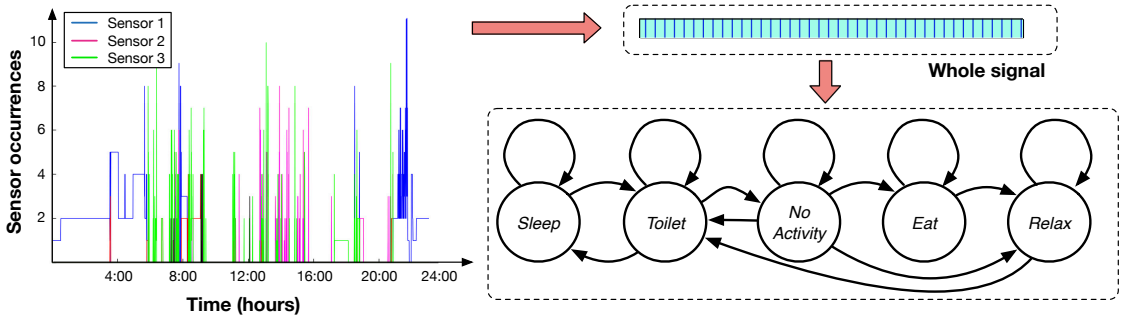


Figure 5.3: Activity recognition with a procedure of windowing without segmentation. Large portions of signal are sent to a HMM. In order to segment the signal, the "non-active" portions are usually represented by a specific state of the HMM.

Activity recognition

The activity recognition is one of the most investigated fields in Smart Home. It differs from presence recognition because it tries to recognize which activity is done without resuming all to an *Active* state. Three different relations between activities can be considered [20]: (i) *sequential*, when the performed activities are contiguous, (ii) *interleaved*, when an activity can be stopped for starting another one and later the previous activity can be finished, (iii) *concurrent*, when a person is carrying out multiple activities at the same time.

Usually in literature the activities are considered as contiguous, therefore HMMs are well suited for the activity recognition task because of the sequential nature of human activities. In its most common configuration, one state per activity is used and the sequence of activities is computed using the Viterbi algorithm [2]. Two approaches are mainly used, depending on the presence of the segmentation step:

- *Windowing with segmentation.* Portions of the "active" signal are detected through the segmentation step. Each window is tested against several GMMs or HMMs representing a different activity, as illustrated in Figure 5.2. The hidden states of HMMs could represent sub-activities, e.g. *Watching television* and *Reading* for the *Relax* activity.

- *Windowing without segmentation.* In this case larger portions of signal, i.e. whole hours or days, are sent to a HMM. This approach is shown in Figure 5.3. Given that the "non-active" and the "active" portions of the signal are not separated through the segmentation, the "non-active" portions are usually represented by a state of the HMM.

The Artificial Intelligence Laboratory of the Washington State University presents several works

in this field. They create CASAS, a system using machine learning technique for discovering recurrent patterns of residents [21]. Data comes by simple environmental sensors such as motion, door, light and temperature sensors. Singla et al. [22] approach the problem of the activity recognition from the temporal point of view and they present their analysis using HMM. In particular they add the timing information by using for each activity duration of each sub-task in an activity. While standard HMM achieves an accuracy rate of 77.3%, HMM including timing information achieves 88.6%. Fang et al. [23] compare three different machine learning techniques for the activity recognition: Naive Bayes classifier (NBC), forward procedure of HMMs and Viterbi algorithm based on HMMs. In particular, they study the relation between feature selection and accuracy rate, finding a strong relation between them. Nazerfard et al. [24] compare the Conditional Random Field (CRF) with HMM on a single dataset obtaining respectively 91% and 82% of recognition for all activities. In another work, Cook et al. [25] use different state-based modeling techniques as NBC, HMM and CRF. They test 11 different datasets using a 3-fold cross validation, obtaining an accuracy rate between 63.3% and 97.3% depending on the dataset. Singla et Cook [26] discuss a more complex scenario, dealing with the interleaved activity recognition. They compare NBC and HMMs achieving respectively an accuracy rate of 61% and 71%.

Van Kasteren et al. [27] realize a system for the activity recognition using HMMs. The dataset is taken by a real home with 14 sensors applied to doors, cupboards, refrigerator and toilet flush. The activities classified are: *Leave the house*, *Toileting*, *Showering*, *Sleeping*, *Preparing breakfast*, *Preparing dinner*, *Preparing a beverage* and *Idle*, if no activity is recorded. In their experiment the sensor data is divided in segments of 1 minute. They achieve an accuracy rate of 79.4%. Monekosso and Remagnino present a first work [28] where they perform activity recognition by using a supervised classification. In a second work [29], they try to achieve the same results without labeling data by using unsupervised classification, i.e. K-Means, K-Medoids, EM and a hierarchical agglomerative method. Data comes from temperature, motion, pressure, window, door, light sensors and smoke detector. The classified activities are: *Sleep*, *Bath*, *Out/No activity*, *Entry/Exit*, *Cook*, *Eat*, *Work* and *Relax*. HMMs therefore model the behavior using the information coming from the supervised or the unsupervised classification. As expected, the supervised classifiers perform better than the unsupervised classifiers. Finally they compare the log-probability of new sequences with those representing "normal" behaviors in order to detect anomalous behaviors. Arora [20] uses a dataset containing motion, infrared, pressure, light, acceleration, temperature and switch sensors. The classified activities are: *Taking medication*, *Toileting*, *Answering phone*, *Sleeping*, *Watching TV*, *Reading books*, *Choosing outfit* and *Cleaning*. Cameras record the Smart Home environment for having access to the ground truth. The results of the activity recognition task are evaluated when using a HMM with a single state per activity and combination of HMM with specific rules, called association rules and temporal rules. He achieves an accuracy rate of 77.2% when using HMM, 78.2% for HMM with association rules and 80.4% for HMM with temporal rules. Dong B. et al. [30] develop a model sensor-based for the prediction of user behavior for managing the energy consumption and the comfort. They use acoustic, lighting, motion, CO2, temperature, relative humidity sensors. Events such as illumination on/off, motion on/off or temperature increasing are recorded. They use Semi-Markov Model where each state is the combination of events detected on 10 minutes windows. They effectuate a building simulation showing that their system yields a potential energy saving of 30% when compared to other basic HVAC control strategies. Hu et al. [31] present a framework able to perform the recognition of interleaved and concurrent activities. They model the interleaved activities by using skip-chain CRF and they use a correlation graph for modeling the concurrent goals. They test their machine learning method on three different datasets and they attain a better accuracy rate than the baseline methods. Chua et al. [32] explore the spatio-temporal and context relations in a Smart Home. The spatial information is related to *where* the activity is performed, for instance some activities depends on the room in which they are effectuated. The temporal information is related to *when* the activity is performed, for instance some activities are most likely done in a given period of the day. Every activity is described by its own HMM. For testing their system, they suppose that all the activities take place in a room known a priori, the kitchen. Five activities take place: *Prepare toast*, *Prepare cereal*, *Prepare beverage*, *Prepare lunch*, and *Do the laundry* and they achieve 90% of accuracy rate.

An additional problem is represented by the presence of several persons at the same time: this task is usually called *multi-resident activity recognition*. Singla et al. [33] perform a real-time recognition of ADL, being attentive to interrupted and interleaved activities. They collect data from an apartment where two residents perform activities concurrently. They test a single HMM for both the residents

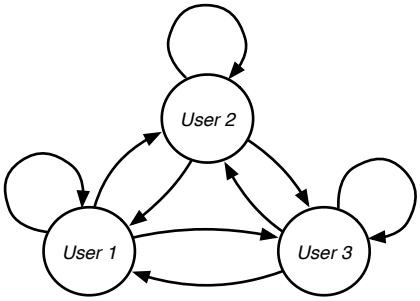


Figure 5.4: Example of a HMM for user recognition. Three users are modeled by three states.

obtaining an accuracy rate of 60.6%. Therefore they use one HMM per resident improving the accuracy rate up to 73.2%. Other more complex modeling solutions able to deal with multi-resident scenarios have been proposed, as in [34, 35].

Resident recognition

When dealing with *multi-resident activity recognition*, models can be used to model the behavior of the person, making possible her or his identification. The information about *who* is performing the activities can be useful. For instance, in medical and assistive applications, it can be used for building the chronology of resident activities. The resident identification is a trivial task when only one resident lives in the Smart Home. In the multi-resident recognition the difficulty of the task is proportional to the number of the residents. Systems using wireless tracking tags can easily identify the resident when the relation between the tag and the resident is known. However this implies the residents to be equipped with personal devices. External devices as cameras and microphones can perform this task typically through the analysis of biometric characteristics, as faces or voices. The resident identification using simple environment sensors is a more difficult problem scarcely studied in literature. Supposing that activities are differently performed by residents, state-based modeling can be used as shown in Figure 5.4.

Crandall et al. [36] use HMMs in order to identify the residents in a Smart Home. They create a model able to identify two different inhabitants. The sensors are passive infrared motion detectors distributed on all the surface, sensors detecting the opening/closing of doors, water flow sensors, etc. The hidden states of the HMM are the possible residents in the datasets. The system has been used on two different datasets and they achieve an identification rate of 96.8% and 93.9%. In a different work they investigate the use of behaviometrics for the identification of inhabitants through the behavior [37]. Their constitutive hypothesis is that simple sensors are able to provide information of different behaviors. They use a HMM where each hidden node represents a single resident: the probability distribution of the features are supposed to be different between residents. By using the Viterbi algorithm they recover the most probable sequence of residents corresponding to the observed events. They test two datasets where they try to recognize the identity of the resident who is performing an activity. They achieve 3.2% and 6.1% of error rate in the identification task. They notice that when the residents are more social they spent more time together in common spaces, causing more difficulties in the identification task. Srinivasan et al. [38] use the height sensors for the biometric identification in a multi-resident Home. These sensors measure the height of the residents: for discriminating two persons with 99% of accuracy rate they need to have at least 7 centimeters difference in height. These sensors are cheap and easy to be installed. They perform two controlled experiments using a lab with 20 subjects and 3 real Smart Homes. Finally they use HMMs on synthetic heights recovered from more than 2000 multi-resident homes. From their simulations they obtain at least 95% identification accuracy in 95% of the multi-resident homes.

Occupation identification

The knowledge about the resident position in a given environment is particularly important for energy saving. For instance, if lights or appliances are on where the activity is not done, they can automat-

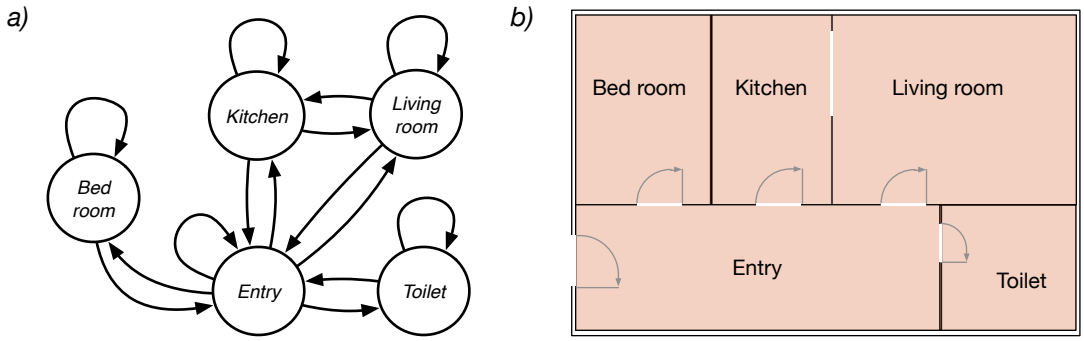


Figure 5.5: Example of an HMM (a) where the topology depends on the home plan (b). We can notice that the states correspond to the rooms, while the transitions correspond to the doors connecting the rooms.

ically be switched off. Specific state-based topologies respecting the a priori knowledge about the sensors position can be used, as illustrated in Figure 5.5. Few works use simple environment sensors, while the large majority uses cameras, as in [39, 40, 41].

De et al. [42] present FindingHuMo, a system able to detect the motion trajectories by using simple binary motion sensors. Their system is able to work real-time in a multi-resident scenario. Paths are discovered through adaptive-HMMs and they propose to use the Crossover Path Disambiguation Algorithm (CPDA) for managing the crossover of the motion trajectories. The adaptive-HMMs choose only a sub-set of "active" states and their neighbors, while CPDA removes the path ambiguities due to the overlaps and crossovers. For evaluating their system, they conduct an experiment where 3 users perform overlapping paths for about 150 seconds. FindingHuMo is compared to a system using a first-order HMM and a second-order HMM: their system perfectly detects the paths, while the two others make some mistakes. Noury et Hadidi present a study on the activity recognition for elderly persons [43] by using presence sensors. They create 7 HMMs corresponding to 7 periods of the day and they systematically use the locations as hidden states. Simulated data of human activities are produced and the quality of the model is evaluated in regard to the correlation with real data. Eventual dangerous situations can be detected and signaled when real and simulated data differs significantly.

5.2.4 Activity duration

As previously seen, HMMs are probably the most popular machine learning technique for the recognition of activities. However, information about duration is not directly taken into account by HMMs, representing a limit for the model. HMMs work efficiently with temporal signals, however the performance tends to degrade when the duration of the temporal segments has a large variability. Activities show long-term temporal dependency that is difficult to deal under the Markov assumption [44]. Information about activity duration helps to identify activities and they have to be included for learning the behavior patterns [45]. This limitation can be bypassed with different strategies based on duration modeling, such as Hidden Semi-Markov Model (HSMM). HSMM defines a variable duration for each state: every state includes a parameter controlling the duration. One observation per state is assumed in HMM while a sequence of observations can be emitted by a state [46].

Some papers deal with the application of HSMM for activity recognition in Smart Home. Duong et al. [44, 47] address the recognition of human activity using HSMM. In particular, they analyze the impact of using different exponential family distributions to model the state duration. They evaluate their system in a Smart kitchen using a multiple camera tracking module for detecting movements and the occupant position. They try to detect the sub-activities performed in the kitchen, where every sub-activity has a particular duration distribution. They analyze contiguous sequences, without the possibility to have gaps between. Van Kasteren [48] applies HSMM to recognize human activities inside three Smart Homes. For each house he performs tests in the bathroom and in the kitchen. Data is collected from sensors placed at doors, appliances, water flush, chairs, etc. Activities are

not necessarily contiguous, and the activity *Other* has been used for the gaps. He compares HSMM with HMM, CRF and Semi-Markov Conditional Random Field (SMCRF), obtaining variable results depending on the house and on the part of the house under analysis. One of the main problems in HSMM is that the Baum-Welch algorithm is not directly applicable and the training phase is slower compared to standard HMMs. An alternative solution is represented by duration models, which will be later explained in details.

5.2.5 Publicly available datasets

Collecting data in Smart Homes is a laborious and time-consuming activity. For this reason several researchers use public available datasets:

- The Artificial Intelligence Laboratory of the Washington State University provides several datasets containing simple environmental sensors. These works have been done in the context of the **CASAS** project [49]. At the time of writing, they give free access to 36 datasets on their website[1]. The datasets differ in the number of residents, type of sensors, duration of recording, availability of annotations and activities.

- MIT presented three **PlaceLab** datasets: PLIA1 (PlaceLab Intensive Activity Dataset 1) [50], PLIA2 [51] and PL Couple (PlaceLab Couple) [52]. In PLIA1 several sensors are placed in an apartment of about 93 m^2: motion, temperature, humidity, light and barometric pressure, electrical current sensors, water flow and gas flow sensors. Several sensors are placed on objects that are often used by people, as chairs, tables and appliances. In some cases residents are wearing accelerators and hearth rate monitors. Cameras and microphones are also placed in strategic positions of the Smart Home. In PLIA2, MIT researchers add information coming from a portable kit of wireless sensors, called MITes. The kit includes six environmental sensors (movement, movement tuned for object-usage-detection, light, temperature, proximity, and current sensing) and five wearable sensors (onbody acceleration, heart rate, ultra-violet radiation exposure, RFID reader wristband, and location beacons). The last dataset consists of 104 annotated hours of a couple living in the instrumented home. More than 900 sensors have been used.

- The University of Amsterdam provides some datasets containing recordings of several sensors and annotations[2]. In particular their provide two datasets: *Ubicomp* and *Transfer Learning*. The first dataset [27] consists of 28 days of data coming from 14 state-change sensors acquired in an apartment having one resident. The dataset contains the annotation of seven different activities. The second dataset [53] contains data coming from three different houses having one resident each. The same set of activities is annotated in the three houses. The houses contain 14, 21 and 23 sensors and data is acquired for respectively 25, 13 and 18 days.

- Other databases are freely available. The **MERLSense** dataset[3] contains a collection of motion sensor data coming from the residential and office buildings Mitsubishi Electric Research Labs (MERL). More than 200 sensors have been recorded for two years. The **Domus** dataset contains data collected from environmental sensors and information about the comfort of the users has been partially annotated. Almost one year of unannotated data is also available. The **LESO-PB** dataset[4] contains more than 6 years of recording at the Solar Energy and Building Physics Laboratory (LESO-PB) building of the EPFL (École Polytechnique Fédérale de Lausanne) by using the EIB/KNX bus. Various sensors and actuators have been recorded, as temperature, presence, lighting, window blind sensors, heating and electric lighting data.

[1] http://ailab.wsu.edu/casas/datasets/
[2] https://sites.google.com/site/tim0306/datasets
[3] https://sites.google.com/a/drwren.com/wmd/details
[4] http://www.wattict.com/web/index.php/databases/14-databases/leso/12-lesodb

5.3 System description

5.3.1 Framework architecture

We realize a framework which is able to recognize and predict human activities by using data collected from simple environment sensors in a Smart Home. We use the predictions on future activities for computing the potential energy savings while preserving the human comfort. We apply the framework on a public dataset (presented in Section 5.3.2) and we show the potential benefits when applying our framework. This framework integrates the feature extraction and three main modules:

1. **Feature extraction**. Data has to be pre-processed in order to make it suitable for the next step, the Activity recognition. The asynchronous observations are transformed in a set of feature vectors equally spaced in time. The feature extraction is presented in Section 5.3.2.

2. **Activity recognition**. The system is able to recognize activities performed by analyzing the processed data. HMMs are used on stream of data coming from the Smart Home. In our framework we test several observation sequences of 24 hours. This module is presented in Section 5.4.

3. **Activity prediction**. By using the information about the activities recognized in the past, the system is able to predict the upcoming activities. Artificial Neural Network (ANN) is used on time windows of the past for this task. This module is presented in Section 5.5.

4. **Energy simulation**. The prediction of the upcoming activities is used by the Energy simulator to estimate the potential energy savings. Information about the building physics and the meteorological conditions are used by this module. This module is presented in Section 5.6.

We compare the activity prediction performances by applying two machine learning techniques directly on features, i.e. HMMs and ANNs. In Figure 5.6 a schematic representation is illustrated.

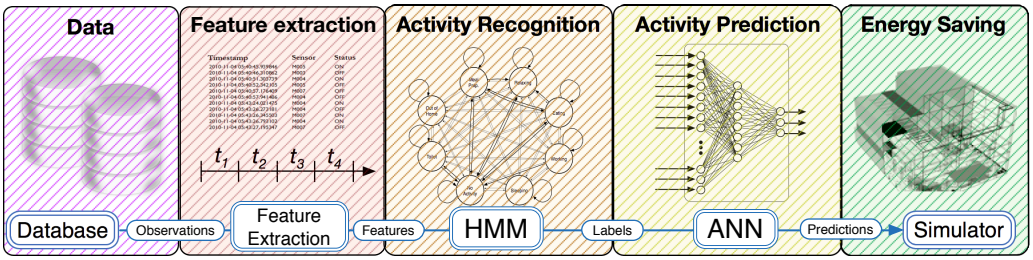


Figure 5.6: A schematic representation of our framework able to analyze data coming from asynchronous sensors, to recognize and to predict the activities and to calculate the potential energy savings. The Activity recognition is performed using HMMs, while the Activity prediction is done with ANNs. A specific software has been applied for computing the energy saving.

This work is one of the few contributions in the literature where there is an evaluation of the machine learning approaches in terms of energy savings. We will compute the energy gains when using a well-know machine learning technique as HMMs for the Activity recognition. The large majority of the publications in this field uses quite complex machine learning techniques often without providing an evaluation of the energy savings.

5.3.2 Data and Feature extraction

We perform the activity recognition and prediction tasks on a labeled CASAS dataset (the *Aruba* dataset), belonging to the WSU CASAS Smart Home project [54]. Raw data in the dataset comes from 31 motion sensors, 4 temperature sensors and 4 door sensors. 11 activities are labeled: *Meal Preparation*, *Relaxing*, *Eating*, *Working*, *Sleeping*, *Dish Washing*, *Bed to Toilet*, *Entering Home*, *Leaving Home*, *Housekeeping*, *Resperate*. The total time of registration is between 7 and 8 months for one

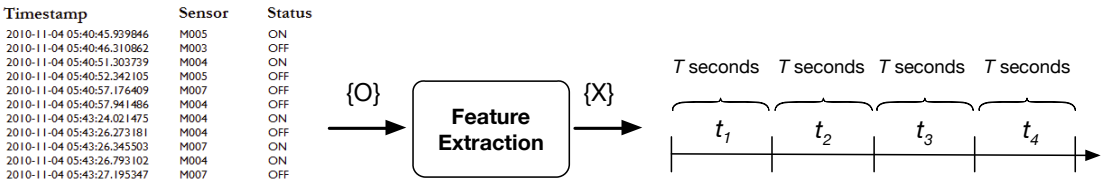


Figure 5.7: Transformation of the set of asynchronous observations $\{O\}$ in a set of features equally spaced in time $\{X\}$.

person living on a daily basis inside the Smart Home and receiving visits on a regular basis. The dataset contains more than 3500000 events.

In our analysis we do not take into account the information coming from the temperature sensors. We process the raw data in order to obtain the observation vectors. All data in the dataset is time stamped, therefore we need to modify the set of asynchronous observations $\{O\}$ in a set of feature vectors equally spaced in time $\{X\}$. We use a moving analysis window for computing the number of occurrences of each sensor in the lapse of time of the window. This process is shown in Figure 5.7. After some pre-tests, the length of the analysis window is chosen equal to 10 seconds. This value is a compromise between granularity (short windows) and quantity of data (long windows). In order to include the information about the time in which an activity is performed, we add to the observation vector the time information computed through a cyclic sinusoidal function. The sinus has a period of 24 hours and its lowest point is set at 2 AM, that is statistically the moment in which the lowest number of occurrences are verified.

We then obtain a sequence of feature vectors $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ where $x_n = [x_{n1}, \ldots, x_{ns}, t_n]$ with x_{ns} equal to the sum of the occurrences of sensors at discrete time n. The feature t_n is a sinusoidal function that depends on the time of the day, having its lowest value at 2 AM and the highest value at 2 PM.

The activities *Leaving Home* and *Entering Home* are grouped together in one activity, *Out of Home*. For the recognition task we remove the activities without an adequate number of data sequences. For this reason we eliminate the activities *Housekeeping*, *Resperate* and *Dish Washing*, reducing their number to 7.

5.4 Activity Recognition

5.4.1 Windowing with segmentation

In this section we analyze the performance of the activity recognition task when using a priori segmented temporal data. We manually perform the segmentation step, simulating a system able to perfectly perform this task. The portions of the signal are classified using models trained for each activity. In particular we compare two methods:

- *Standard GMMs.* Each activity is modeled with a specific GMM.

- *Occupation-based HMMs.* Each activity is modeled with a HMM where the states are related to the expected locations of the activities. In this respect, the signal is separated in two categories, depending on the quantity of occurrences in time: few occurrence variations correspond to a *stable* signal, while many occurrence variations correspond to an *unstable* signal. In practice, the *unstable* portions of the signal are related to the transition areas, e.g. the corridors or the doors, and they are associated to a state labeled as "unstable". On the other hand, the *stable* portions are related to the stationary areas, e.g. the kitchen for cooking or the bedroom for sleeping, and they are associated to a "stable" state. We use the information about the stable/unstable portions in order to label our observation sequences.

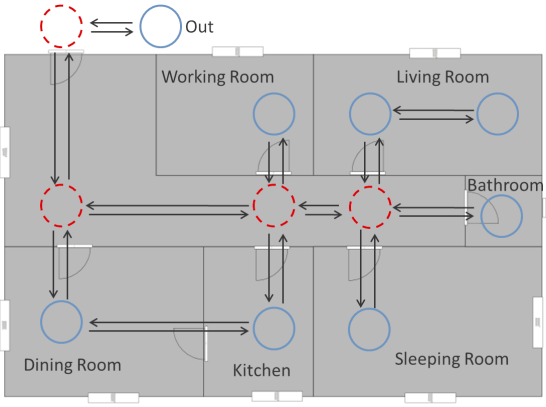


Figure 5.8: Synthetic example of HMM topology where states are related to the spatial areas. Stable and unstable states are represented with a dashed line. The portions of the signal are considered stable or unstable depending on the number of occurrences.

In Figure 5.8, we illustrate an example of the relation between the type of states and the map of the house. For each activity, we search for the relation between sub-activities and the expected locations:

- *Meal Preparation*: prepare the meal (kitchen) and set the table (dining room).
- *Relaxing*: arrive to the couch (living room part A), stay on the couch (living room part B) and leave the couch (living room part A).
- *Eating*: cook (kitchen) and eat (dining room).
- *Working*: go to work (corridor), work (office) and leave the office (corridor).
- *Sleeping*: go to bed (corridor), sleep (bedroom), and stand up (corridor).
- *Out of Home*: leave home (door), stay out (out) and enter home (door).
- *Toilet*: stand up (corridor), going to toilet (toilet) and coming back (corridor).

We relate the kitchen, the dining room, the living room part B, the office, the bedroom, out of home and the toilet to stable states, the others to unstable states. We observe that all activities can be represented by one between two different topologies, shown in Figure 5.9. The topology represented in Figure 5.9a corresponds to a simple ergodic configuration composed of two states. It is used for the activities *Meal Preparation* and *Eating*. In this case, the states are associated to the same areas, the kitchen and the dining room. The topology of Figure 5.9b corresponds to a three states model. It is used for the other five activities: *Relaxing, Working, Sleeping, Out of Home, Toilet*. In this case, the

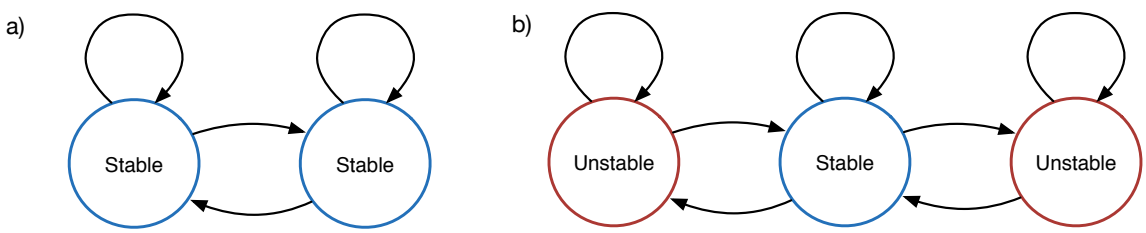


Figure 5.9: The first topology a) is used for *Meal Preparation* and *Eating*, and the second one b) is used for *Relaxing, Working, Sleeping, Out of Home, Toilet*.

	Meal Prep.	Relaxing	Eating	Working	Sleeping	Toilet	Out Home
Meal Prep.	1	0	0	0	0	0	0
Relaxing	0	.96	0	0	0	0	.04
Eating	.02	.02	.96	0	0	0	0
Working	0	0	0	1	0	0	0
Sleeping	0	0	0	0	1	0	0
Toilet	0	0	0	0	0	1	0
Out of Home	.01	.01	.01	.04	.02	0	.91

Table 5.1: Confusion matrix obtained when modeling each activity with a GMM.

	Meal Prep.	Relaxing	Eating	Working	Sleeping	Toilet	Out Home
Meal Prep.	1	0	0	0	0	0	0
Relaxing	0	1	0	0	0	0	0
Eating	.02	.01	.97	0	0	0	0
Working	0	0	0	1	0	0	0
Sleeping	0	0	0	.01	.99	0	0
Toilet	0	0	0	0	0	1	0
Out of Home	.01	.01	.01	.02	0	0	.95

Table 5.2: Confusion matrix obtained when using the occupation-based HMM.

resident starts the activity arriving to the stable area through a transition zone and ends the activity after leaving the location. For this reason, the first state is unstable, the second is stable and the third unstable again.

We train the system and compute the accuracy rate by using a leave-one out procedure. By using a standard GMM, namely when activities are represented by only one state, we obtain an accuracy rate of 97.7%. In Table 5.1 we report the confusion matrix. When using the occupation-based HMM, we obtain an accuracy rate of 98.9%. Table 5.2 shows the confusion matrix. In the latter case we obtain up to 50% error reduction over standard GMMs. The worst class is *Out of home* with 95% and 91% accuracy rate respectively for the proposed system and the classical approach. This is probably due to the fact that no sensor is activated during this activity and consequently, the system is very sensitive to the observation data in the first and last part of the sequence. In both approaches, the activity *Toilet* has no false positive nor false negative.

As previously said, the main hypothesis of such approach is that signal has to be previously segmented in order to be processed, which is an optimistic situation. In the next section we provide

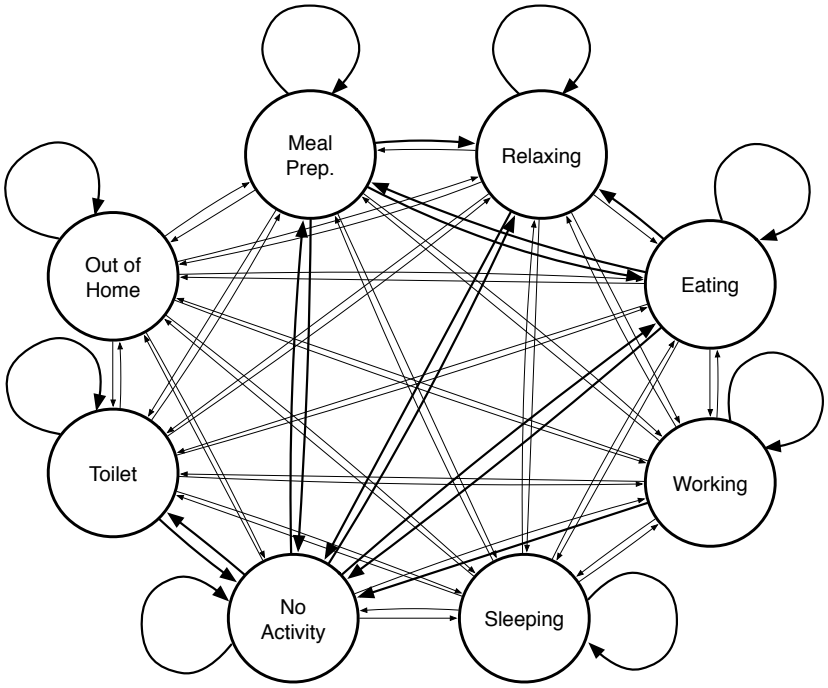


Figure 5.10: The ergodic HMM used for the recognition of 8 activities: *Relaxing*, *Eating*, *Working*, *Sleeping*, *No activity*, *Toilet*, *Out of home* and *Meal preparation*.

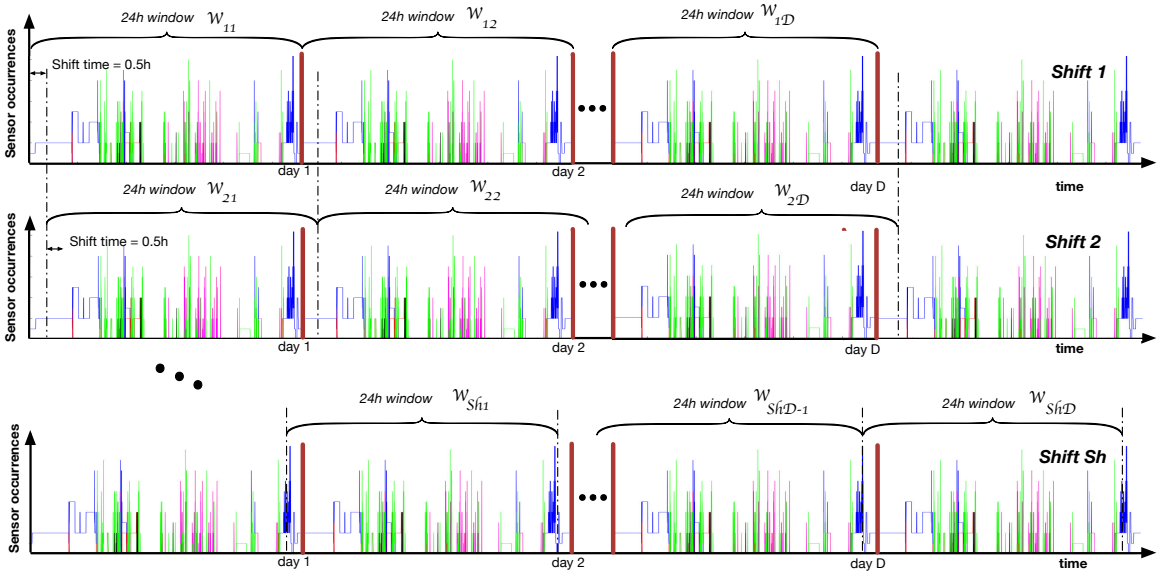


Figure 5.11: Windowing of the signal. We generate different sets of windows considering a different shift, equal to 30 minutes. The graphs of different colors (blue, green and pink) represent trend of the occurrences for different sensors.

a different approach able to automatically segment data and perform Activity recognition.

5.4.2 Windowing without segmentation

In this implementation, we build a HMM able to model portions of the signal without any a priori segmentation. Each state of the HMM represents a different activity. One state represents the *No Activity* case, that occurs when all the other activities are not present. Given that we do not have any a priori information about the activity sequence, we configure the states as they are ergodically connected, as illustrated in Figure 5.10. Through the Viterbi algorithm, we obtain the most probable sequence of hidden states for every observation. The accuracy rate is computed by comparing the most probable sequence of states with the ground truth.

In this type of analysis, given that we do not use the segmentation, we need to artificially decide the length of our observation sequences. We choose the length of the observation sequences to be 24 hours, in such a way we can cyclically apply our model across different days. The shift time between two consecutive windows is 30 minutes. The set of all the windows $\mathcal{W}$ can be seen as the totality of the windows with different shifts through different days:

$$\mathcal{W} = \{\mathcal{W}_{11}, \mathcal{W}_{12} \ldots \mathcal{W}_{1D}, \mathcal{W}_{21}, \mathcal{W}_{22}, \ldots, \mathcal{W}_{2D}, \ldots, \mathcal{W}_{Sh1}, \mathcal{W}_{Sh2}, \ldots, \mathcal{W}_{ShD}\} \tag{5.1}$$

where $D = 220$ is the total number of days and $SH = 48$ is the total number of windows into a day. This procedure is depicted in Figure 5.11. We divide the set of daily observations in three parts: train ($\mathcal{W}_{tr}$), validation ($\mathcal{W}_{val}$) and test ($\mathcal{W}_{ts}$) sets, which represent respectively 50%, 10% and 40%. In practice, we use the first 110 days for the training, the next 22 days for the validation and the rest for the test. The validation set is necessary for tuning the HMM parameters and the final accuracy rate is estimated on the test set. In order to train our system, the best choice would be to use all data contained in $\mathcal{W}_{tr}$. However, given the large quantity of data this choice is impracticable. Therefore we propose three alternative solutions in order to train our model:

- *Random selection training.* We randomly select a sub-set of $\mathcal{W}_{tr}$ for training our model, searching a compromise between quantity of data and execution time. In our selection we pay attention to select at least one observation per day among the 48 available windows. Given that the selection is randomly performed, we decide to repeat the tests 5 times and average the results.

- *Specific shift training.* We train a HMM using windows in the training set with a specific shift. We take the shift corresponding at 2 AM.
- *Complete Shift training.* We train 48 different HMMs, every one using windows in the training set with the same shift.

In Table 5.3 we resume the details of the different trainings.

	Number of models	Number of windows
Random Selection training	5 HMMs	$5 \cdot 110 = 550$
Specific Shift training	1 HMM	$1 \cdot 110 = 110$
Complete Shift training	48 HMMs	$48 \cdot 110 = 5280$

Table 5.3: Resume of the training methods. We compare three trainings: *Random Selection*, *Specific Shift* and *Complete Shift*.

5.4.3 Minimum duration modeling

In this thesis we apply a conceptually simple variation of HMMs that exploits the potentialities of duration modeling by repeating the states generated after the training phase. Such method has been used also in other fields, e.g. speech recognition [55] and text recognition [56, 57]. However, this method has never been proposed, to the best of our knowledge, for human activity recognition.

The minimum duration modeling is a method exploiting information about the duration of every state. It mainly consists in forcing the algorithm to remain in a certain state for a specific amount of time by modifying the model topology. The topological alteration is produced after the training phase and before the test phase. In practice a specific state is substituted with a copy of the same state replicated several times (*chain*). The rules below are followed:

- The first state of the chain is connected as output with the last states of the other chains. The first state is connected as input to the second state of the chain.
- The central states are connected as input only to the next ones in the chain. Self transitions are not admitted.
- The last state of the chain is the only one admitting the self-transition. It is connected as input to the first state of other chains.

The transition matrix of the new model is derived from the original by maintaining the same probabilities between the first and last states of the chains and adding the probability (100%) between

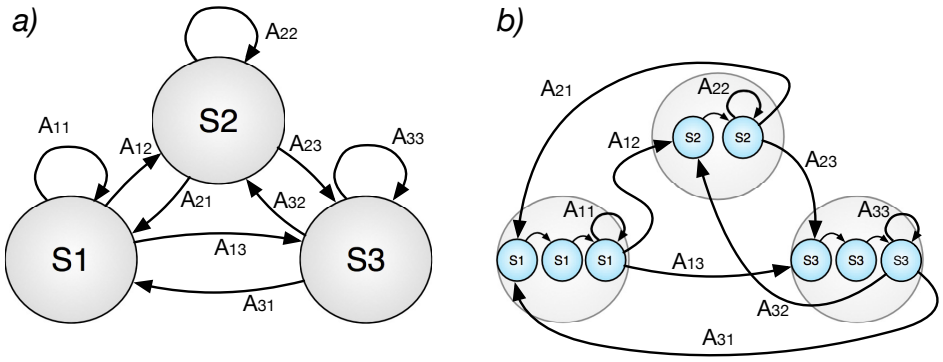


Figure 5.12: An ergodic model with three states (a) is modified into a model with a new topology (b). In this case we have three repetitions of the first state, two for the second and three for the third.

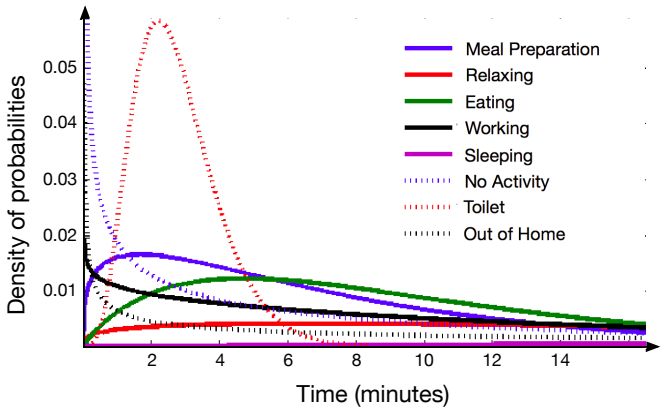


Figure 5.13: Approximation of histograms of the activity duration by using gamma distributions.

the central states in the chain. An example is shown in Figure 5.12. An ergodic model with 3 states (a) is modified into a model with a new topology (b). For sake of simplicity, we consider chains with few repetitions (3 for the first state, 2 for the second and 3 for the third). The new topology presents 8 states and the chains are connected only through their first and last states.

Exploiting the duration of activities can improve the accuracy rate of the classification and can provide important information on abnormalities. For instance, if a person remains in one location for more time than usual, a problem might have occurred and eventually an alert could be activated. In particular the elder care domain could benefit of such system [44].

As first step we compute the histograms of duration for each activity. In Figure 5.13 we illustrate an approximation of the activity histograms by using the gamma distribution, often used in literature for this kind of task. The figure reports the duration of 8 activities, including the *No Activity* case. The duration distribution of some activities, e.g. *Sleeping*, is not clearly shown given the small scale of the figure.

For computing the lengths of the chains of states two strategies can be used:

1. **Mean-based length modeling**. The minimum duration values are inferred at training time by accumulating in histograms the number of times the self-loop transition is visited for all states. The average number of self-loop transitions μ_i is computed from the histograms. The minimum duration values are set to the half of the average value ($\mu_i/2$) [55].

2. **Quantile length modeling**. As explained in [58], the length of a chain is set to a specific quantile (Q) of the corresponding activity duration histogram. We observe the trend of the accuracy rate computed on the validation set when varying the quantile, looking for the quantile yielding to the best accuracy rate. The *Mean-based length modeling* strategy is a specific case of the *Quantile length modeling*, i.e. $Q = 25$ is the best approximation of $\mu_i/2$. For this reason in the work below we directly use the *Quantile length modeling*.

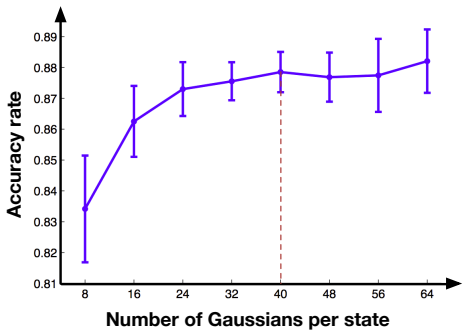
We use the tuning set for finding the best parameters. Given that we use as probability density function the mixture of Gaussians, we tune the number of Gaussians per state (K). For every training, we illustrate the trend of the accuracy rate on the validation set when increasing the number of Gaussians. We start with $K = 8$ and we iteratively add 8 Gaussians at each step until $K = 64$. Given that we use the *Quantile length modeling*, we tune the quantile (Q). Knowing that $Q = 25$ is a good approximation of the quantile, we search around this value. We start with $q = 17.5$ up to 40 with a step of 2.5.

We report below a summary of the tuning values depending on the type of training:

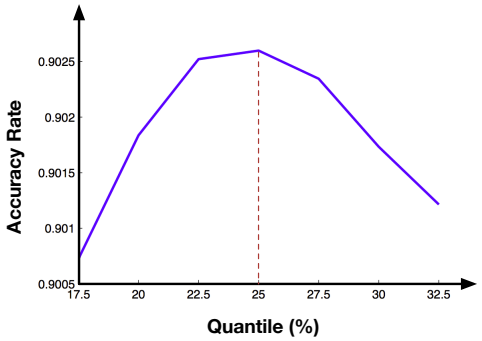
- **Random selection training**

  In Figure 5.14a we report the trend of the average of the accuracy rate when increasing the number of Gaussians. Given that the average is computed on 5 random repetitions, we also report the standard deviations. When using 40 Gaussians per state, the accuracy rate seems to be saturated. Even if $K = 64$ performs slightly better, we decide to take $K = 40$ in the

further analysis considering the great difference in training time. In Figure 5.14b we illustrate the accuracy rate when varying the quantile. $Q = 25$ is the best case and it improves the accuracy rate from 87.9% to 90.3%.



(a) Trend of the accuracy rate on the validation set when varying the number of Gaussians per state.

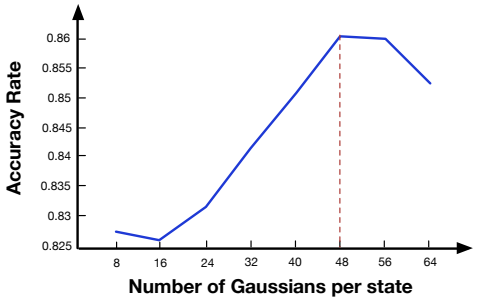


(b) Trend of the accuracy rate on the validation set when varying the quantile.

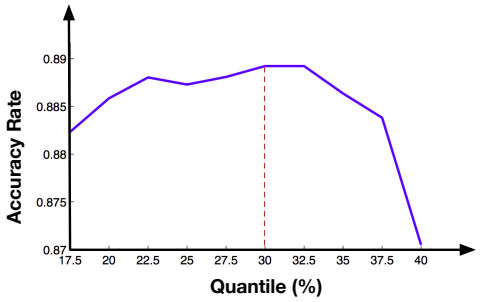
Figure 5.14: Representation of the accuracy rate for the *Random selection training*.

- **Specific shift training**

  In Figure 5.15a we report the trend of the accuracy rate when increasing the number of Gaussians. We choose $K = 48$ as it saturates the accuracy rate. In Figure 5.15b we illustrate the accuracy rate when varying the quantile. $Q = 30$ yields the best results and the accuracy rate is improved from 86.1% to 88.9%.



(a) Trend of the accuracy rate on the validation set when varying the number of Gaussians per state.

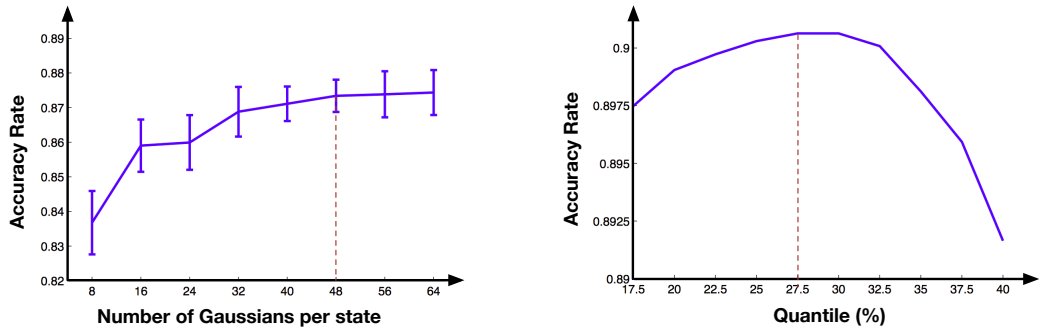


(b) Trend of the accuracy rate on the validation set when varying the quantile.

Figure 5.15: Representation of the accuracy rate for the *Specific shift training*.

- **Complete Shift training**

  In Figure 5.16a we report the trend of the average of the accuracy rate when increasing the number of Gaussians. The average is computed from the 48 different HMMs. In figure we also report the standard deviations. As the previous case, $K = 48$ seems to saturate the results. In Figure 5.16b we illustrate the accuracy rate when varying the quantile. $Q = 27.5$ yields the best results and the accuracy rate is improved from 87.3% to 90.1%. In the Figure 5.17 we report the trend of the accuracy rate on the validation set when varying the quantile and the shift. We notice a smooth trend in the direction of the quantile, however we do not notice a similar behavior along the shift. HMMs are trained on specific shift, therefore HMMs are specialized on a certain time of the day. As a consequence, the performance of HMMs depends on the regularity of the distribution of the activities during the day.



(a) Trend of the accuracy rate on the validation set when varying the number of Gaussians per state.

(b) Trend of the accuracy rate on the validation set when varying the quantile.

Figure 5.16: Representation of the accuracy rate for the *Complete shift training*.

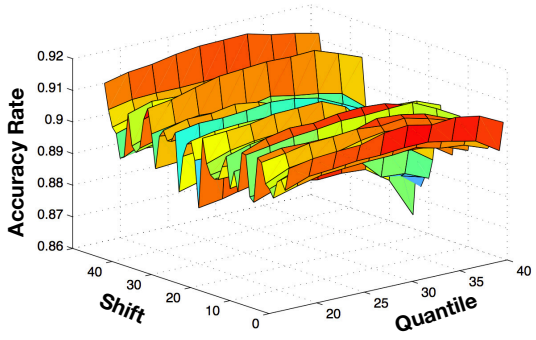


Figure 5.17: Accuracy rate on the validation set when varying the quantile and the shift.

After the tuning phase, we use the best number of Gaussians per state and quantile on the test set for every type of training. In Table 5.4 we report the summary of the Activity recognition results on the validation and test set for all the types of training.

	Validation set		Test set	
	tuning K	tuning Q	K tuned	Q tuned
Random Selection training	87.9 ($K = 40$)	90.3 ($Q = 25$)	84.2	86.7
Specific Shift training	86.1 ($K = 48$)	88.9 ($Q = 30$)	84.6	86.7
Complete Shift training	87.3 ($K = 48$)	90.1 ($Q = 27.5$)	84.0	86.4

Table 5.4: Resume of the Activity recognition results. We compare three trainings: *Random Selection*, *Specific Shift* and *Complete Shift*. We systematically compute the best values for the validation set and accuracy rates when using the tuned values for the test set.

We can observe two interesting facts:

- The minimum duration modeling always increases the performances, independently on the set (validation or test) and the training methods (*Random Selection*, *Specific Shift* or *Complete Shift*). In particular the improvement always ranges between 2% and 3% of accuracy rate.
- The three training methods provides very similar results. Only the *Specific Shift* method provides slightly worse results on the validation set, however this difference does not appear on the test set. The *Complete Shift* training provides slightly worse results on the test set and maybe this is due to the fact that 30 minutes is too short for capturing daily routines.

	Meal Prep.	Relaxing	Eating	Working	Sleeping	No Activity	Toilet	Out Home
Meal Prep.	.89	.01	.04	0	0	.05	0	0
Relaxing	.01	.93	0	0	0	.05	0	0
Eating	.06	.01	.87	0	0	.06	0	0
Working	0	0	0	1	0	0	0	0
Sleeping	.01	0	0	0	.98	.01	0	0
No Activity	.12	.09	.11	.03	.05	.39	.17	.03
Toilet	0	0	0	0	0	.01	.99	0
Out of Home	.01	0	0	.02	0	.02	0	.94

Table 5.5: Confusion matrix when applying standard HMMs using the *Specific Shift* training.

	Meal Prep.	Relaxing	Eating	Working	Sleeping	No Activity	Toilet	Out Home
Meal Prep.	.93	.01	.01	0	0	.05	0	0
Relaxing	.01	.93	0	0	0	.06	0	0
Eating	.08	0	.85	0	0	.07	0	0
Working	0	0	0	.99	0	.01	0	0
Sleeping	.01	0	0	0	.95	.04	0	0
No Activity	.12	.07	.07	.01	.03	.56	.13	.01
Toilet	0	0	0	0	.48	.02	.5	0
Out of Home	.02	0	0	.02	0	.02	0	.93

Table 5.6: Confusion matrix when applying HMM with quantile length modelling using the *Specific Shift* training.

We decide to focus on the *Specific Shift* training given that it gives the best performances on the test set while requiring the smaller number of HMMs to be trained. In Table 5.5 and 5.6 we report the confusion matrices for the standard and duration modeling analysis for the *Specific Shift* training. In the first case all the states are quite well recognized with the exception of the *No Activity* state which is often misclassified with all the other states. This is mainly due to the fact that several small segments of *No Activity* state are misclassified. In fact, motion sensors are often very sensitive and therefore they are important source of noise especially during the *No Activity*. In the second case single states are generally better recognized, with the exception of the *Bed to Toilet* activity, for which the result drops to 50%. However this activity is scarcely present, so the impact on the accuracy rate is weak. The *No Activity* recognition rate when using the minimum duration modeling increases from 39% to 56%. We observe a significant improvement of the segmentation capability of the system when including the information on activity duration. In practice we improve the system capability to detect if an activity is performed or not. This improvement is mainly due to the fact that with minimum duration modeling we limit the number of small segments misclassified as other activities.

5.5 Activity Prediction

As shown in Figure 5.6, we propose a system that uses HMMs for the activity recognition and an ANN for the activity prediction. The ANN is fed with the activity labels generated by the HMM. In order to analyze the activity prediction performances, we use HMMs and an ANN directly on features for forecasting the future activities, skipping the Activity recognition step. In this section we will provide details on the three prediction systems: (i) HMMs on features, (ii) ANN on features and (iii) ANN on labels coming from the Activity recognition.

5.5.1 Pre-processing

Activity simplification

The goal of the Activity Prediction module is to forecast which activity will be done in the future at few hours of distance. The information coming from this module is transmitted to the Energy Simulator, able to estimate the energy saving by using the predicted values. Given that the prediction task is quite difficult, we decide to simplify this task while providing useful results for the Energy Simulator module. We group the 8 activities in 3 clusters depending on their impact on the energy consumption. We group together *Meal Preparation*, *Relaxing*, *Eating*, *Working*, *Bed to Toilet* and *No activity* in an unique activity, that we call *Activity at Home*. The other two activities, i.e. *Sleeping* and *Going out of Home*, have been left as is. For sake of simplicity, we will refer to *Going out of Home* as *Activity 1*, to *Sleeping* as *Activity 2* and to *Activity at Home* as *Activity 3*.

Signal Windowing

In order to make predictions, we use the samples of the past as input and those of the future as output. In particular we consider two windows in the past: a longer one with a lower sampling frequency (called *Long Window*) and a shorter one with a higher sampling frequency (called *Short Window*). Several parameters have been used for this analysis:

- F_{long} is the distance in time between two consecutive samples of the *Long Window*.
- T_{long} is the time length of the *Long Window*.
- F_{short} is the distance in time between two consecutive samples of the *Short Window*.
- T_{short} is the time length of the *Short Window*.
- F_{shift} is the distance in time between two consecutive predictions.
- T_{fut} is the distance in time of the forecasting.

In the Figure 5.18 we illustrate the signal windowing and some of the parameters.

The machine learning model is fed with the information of the *Long Window* and *Short Window* samples. In addition we add information about the day of the week and the time of day. This information could potentially be interesting given that several activities will occur on a regular daily or weekly basis [59].

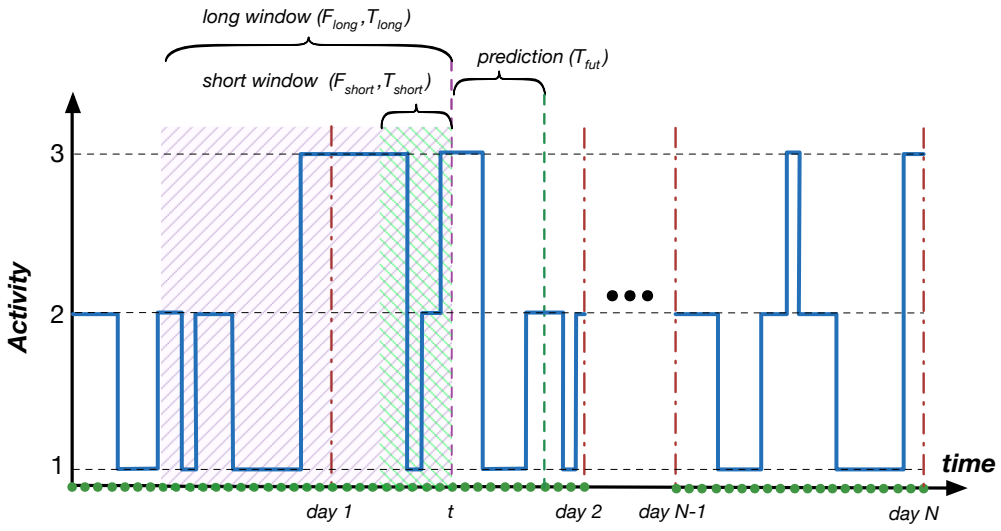


Figure 5.18: Two windows are extracted from the past samples: (i) the *Long Window* with the T_{long} and F_{long} parameters and (ii) the *Short Window* with the T_{short} and F_{short} parameters. Moreover the distance in time of the forecasting T_{fut} is represented.

We decide to fix some of these parameters after some pre-tests, i.e. F_{long} is equal to 5 minutes, F_{short} is equal to 10 seconds and F_{shift} is equal to 30 minutes. We perform 6 different tests, corresponding to 6 different T_{fut}: 1 hour, 2 hours, 3 hours, 4 hours, 5 hours and 6 hours. For every T_{fut} we systematically tune the T_{long} and T_{short} parameters. T_{long} has been set between 1 and 24 hours, while T_{short} has been set between 5 and 60 minutes.

5.5.2 HMMs on features

As first step, we use two groups of HMMs directly on features in order to predict future activities. More specifically, we use two HMMs, as we consider two windows of different length in the past: a group of HMMs is dedicated to the classification of the *Short Window*, while another is dedicated

to the *Long Window*. Each group of HMMs contains three HMMs, one per activity to be predicted. The HMMs are ergodic with three states, one per activity. The windows are labeled depending on the activity predicted. When taking the decision about the winning class, both groups of HMMs are considered: the log-likelihoods of the two windows are added together. However, the log-likelihoods are tributary of the number of samples in the observations, that is different between the two windows. As a consequence, before adding the log-likelihoods, they are weighted for the length of the windows, i.e. the final score is divided by the window length. The main procedure is illustrated in Figure 5.19.

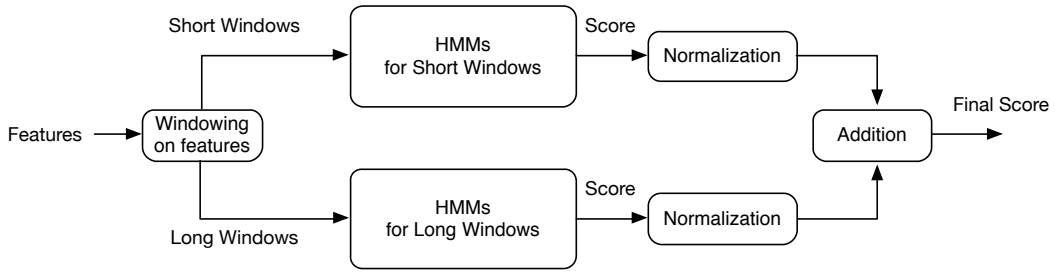


Figure 5.19: Our procedure for HMMs on features. The features are windowed according to the *Short Window* and *Long Window* procedure. The time series generated are used for training two groups of HMMs, one for type of windows. The labels depend on the activity predicted. During the testing, the scores are divided for the length of the windows before adding them.

Tuning

As previously said, the length of the Short Window (T_{short}) varies between 5 and 60 minutes using a fixed sampling frequency (F_{short}) of 10 seconds, while the Long Window (T_{long}) varies between 1 and 24 hours using a sampling frequency (F_{long}) of 5 minutes. We use a validation set for estimating the best length of both windows. In the test phase we use these values for evaluating the goodness of the prediction on the test set. Six different predictions have been analyzed, namely $T_{fut} = 1, 2, 3, 4, 5, 6$ hours.

For the Short Window we notice that the accuracy rate does not sensibly change when varying T_{short}, showing that the length of the Short Window does not have a relevant impact. The opposite occurs for the length of the Long Window, showing to be an important parameter for the forecasting. In Figure 5.20a we report the trend of the overall accuracy rates when varying the length of the Long Window for a fixed Short Window length, i.e. $T_{short} = 5$ minutes. In particular we notice that the accuracy rate decreases when increasing the length of the Long Window. This result shows the limitation of HMMs when they are used for the prediction task.

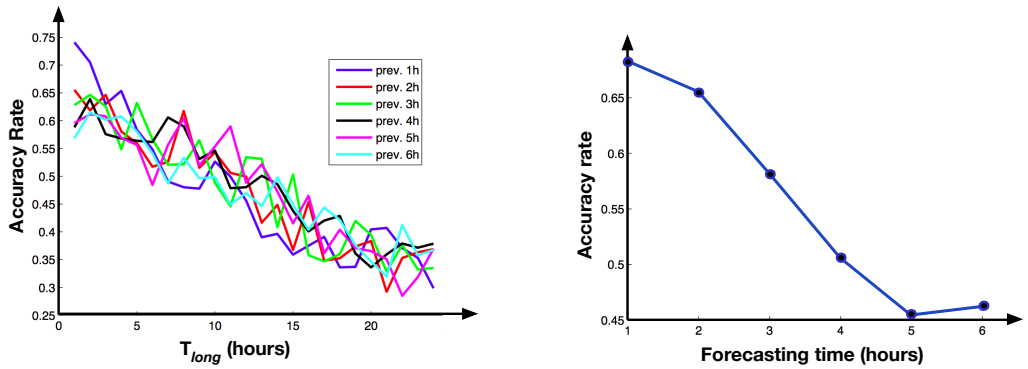
Testing

In the Figure 5.20b we report the trend of the accuracy rate while increasing the forecasting time. The results are obtained on the test set by using the best combination of Short and Long Window lengths after the tuning phase. We observe that the results appear very mitigated when T_{fut} is greater than 3 hours, showing that the accuracy rate attains values below 50% for a 3-class problem.

5.5.3 ANN on features

In this test we use a machine learning algorithm more suitable for this task, the Artificial Neural Network (ANN) directly on features. One of the major application areas of ANNs is forecasting [60]. ANNs are typically used to estimate or approximate functions between inputs (past samples) and outputs (future samples).

ANN is inspired by the animal nervous system, in particular the brain. Basically the brain is made of several millions of nerve cells, called neurons, which communicate between themselves through electrical and chemical signals. Each neuron receives several inputs x_i having a weight w_i. All the



(a) Overall accuracy rate when increasing the length of the long window on the validation set.

(b) The trend of the accuracy rate when increasing T_{fut} by using HMM on the test set.

Figure 5.20: Evolution of the accuracy rate for the HMM on features.

weighted inputs are summed and they are passed through the neural activation (or transfer) function $f(a)$. The output g of the activation function is therefore transmitted to the other neurons. The output can be written as a function of the inputs:

$$g = f(a) = f(\sum_i w_i x_i) \tag{5.2}$$

ANN have three types of neurons:

- *Input neurons* receive the information from the external environment. All together they form the input layer.
- *Hidden neurons* compute the relation between the input and the outputs. They can be arranged on different layers, called hidden layers.
- *Output neurons* send the information to the external environment. All together they form the output layer.

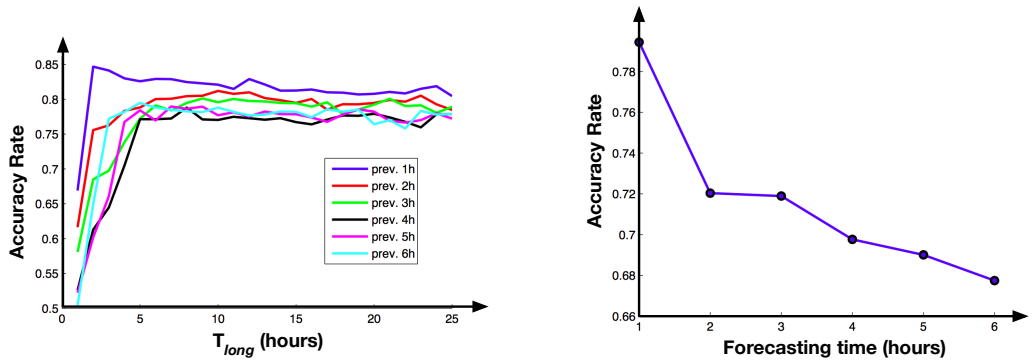
The activation function has to be differentiable for the training procedure. The most used activation functions are the *sigmoid* and the *tanh* functions (in this work we use the *sigmoid*). The most used neural network model is the *multi layer perceptron* (MLP), that is also used in this work. With feedforward MLPs, neurons are connected from the input to the output without any possible feedback and loop. MLPs have at least three layers of neurons: one for the input layer, one for the output layer and at least one for the hidden layer. The network is trained by updating the weights of the neurons. We train the MLP by using the back-propagation algorithm, that iteratively updates the weight of the neurons in order to minimize the errors via a gradient descent procedure. More information can be found in [61].

In order to predict the future activities, we feed an ANN with the samples of the Short and the Long Windows. Every sample has a number of dimensions D equal to the number of sensors used in the environment plus a dimension for the time. In a Neural Network, a single sample can be represented by a number of input neurons equal to the number of its dimensions. As a consequence, the total number of input neurons N_i is equal to:

$$N_i = (N_{short} + N_{long}) \cdot (D + 1) \tag{5.3}$$

Where N_{short} and N_{long} are respectively the number of samples in the Short and Long Windows.

The main problem of such approach is that the number of input neurons is directly proportional to the number of dimensions, and therefore to the number of sensors in the environment. As a consequence, when considering a large number of sensors, the network is quite complex, requiring a large quantity of resources and training time. In the worst case, i.e. for the longest windows, we have more than 20000 input neurons.



(a) Prediction using ANN fed with features on the validation set.

(b) Overall accuracy rate when increasing the forecasting time on the test set.

Figure 5.21: Evolution of the accuracy rate for the ANN on features.

Tuning

As in the previous case, we vary T_{short} between 0 and 60 minutes and T_{long} between 0 and 24 hours while the sampling frequencies are constant. We use a validation set for estimating the best length of the two windows. Six different predictions are made, from 1 hour to 6 hours with a step of 1 hour.

As for the HMM, we do not notice any important change when varying the length of the Short Window, while the opposite occurs when considering the Long Window. In Figure 5.21a we report the trend of the accuracy rate when varying the length of the Long Window for a fixed Short Window length, i.e. $T_{short} = 5$ minutes. In this case the accuracy rate increases when increasing the length of the Long Window. All the curves saturate when using few hours in the past. However, depending on the prediction T_{fut}, different values of T_{long} seem to saturate the accuracy rate.

Testing

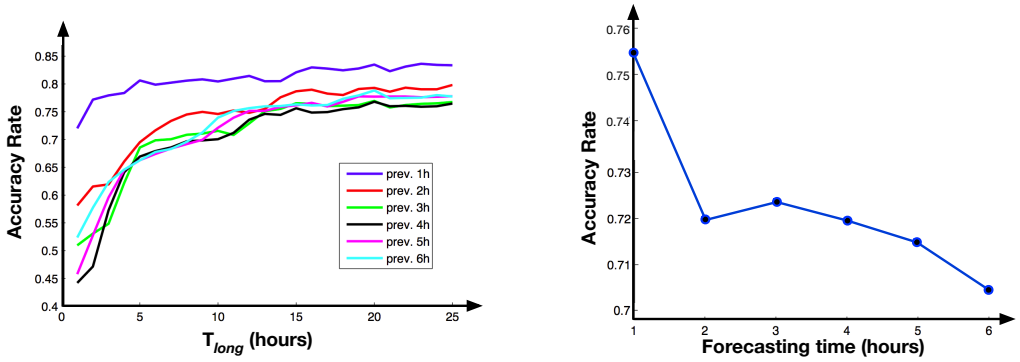
During the tuning phase we compute the best length of the short and long window for every forecasting time. During the testing phase we use these values for computing the accuracy rate on the test set. The results appear quite noisy due to the random initialization of the weights. In Figure 5.21b the trend of the accuracy rate is illustrated. As expected, the accuracy rate tends to degrade when increasing the forecasting time. In particular for a prediction of 1 hour, the accuracy rate is about 79%, while for the other forecasting times it ranges between 68-72%.

5.5.4 ANN on labels

As previously said, HMMs provide unsatisfactory results for the prevision task, especially after 3 hours of prediction. When dealing with ANN, better results are provided in term of accuracy rate, however it poorly scales with the number of sensors. Our hybrid framework is able to improve these two aspects.

As shown in the previous section, we use the HMM in order to perform the *Activity Recognition* step. Through the Viterbi algorithm we are able to retrieve the labeling of the activities done in the past. Simultaneously we train an ANN by using the ground truth of the activities in order to find the relation between the past and future activities. The ANN is therefore fed with labels coming from the *Activity Recognition* and trained to forecast future activities.

The difference with the previous system is that, in this case, the features are processed by the HMM for the Activity recognition and the ANN processes the labels generated by the HMM. Every sample has a number of dimensions equal to the number of activities N_{act} plus a dimension for the time. N_{act} in the specific case is equal to 3. Each sample is therefore represented by its relative activity and, as commonly done when dealing with discrete values, we set the neuron associated to the relative activity to "1", while the others are set to "0". As a consequence, the total number of



(a) Overall accuracy rate when increasing the length of the long window on the validation set.

(b) Overall accuracy rate when increasing the forecast prediction time for the HMM-ANN case on the test set.

Figure 5.22: Evolution of the accuracy rate for the ANN on labels.

input neurons N_i is equal to:

$$N_i = (N_{short} + N_{long}) \cdot (N_{act} + 1) \tag{5.4}$$

As shown in the previous formula, the number of input neurons does not depend on the number of sensors, as in the previous case, but on the number of activities. Given that in our case only three activities have been considered, the network is smaller and very fast to train.

Tuning

As previously done, we vary T_{short} between 0 and 60 minutes and T_{long} between 0 and 24 hours while the sampling frequencies are constant. We use a validation set for estimating the best length of the two windows. Six different predictions are made, from 1 hour to 6 hours with a step of 1 hour.

During the tuning phase, we use the ground truth of the labels in order to reduce potential noises between past and future samples. As for the HMM and the ANN, the length of the Short Window has a poor impact on the accuracy rate, while the opposite occurs when considering the Long Window. In Figure 5.22a we report the trend of the accuracy rate when varying the length of the Long Window for a fixed Short Window length (i.e. $T_{short} = 5$ minutes). The accuracy rate increases when increasing the length of the Long Window, however the curves saturate when using several hours in the past. However, depending on the prediction T_{fut}, different values of T_{long} saturate the accuracy rate.

Testing

Figure 5.22b illustrates the evolution of the accuracy rate when increasing the parameter T_{fut}. As expected, by increasing the forecasting time the trend of the accuracy rate appears to degrade. As in the previous cases, the best accuracy rate is provided by the prediction of 1 hour (about 75.5%). For the other forecasting times the accuracy rate ranges between 70-72.5%.

5.5.5 Results

The ANN applied to the labels provides better results than the HMM on features in term of accuracy rate. The ANN applied to the labels provides slightly worse results compared to the ANN on features for short prediction values, while the opposite occurs for longer forecast times. Morever, the ANN on labels appears more scalable than the ANN on features when increasing the number of sensors. For this reasons, we decide use the ANN on features for the next step. The sequences of predictions are sent to the next module, the *Energy Simulator*, which simulates the application of a standard HVAC system and computes the possible energy saving measures depending on the building characteristics. More information is provided in the next section.

Construction elements	Room in building with heavy construction	Room in a light-weight building
Floor area	16 m^2	
Room height	2.8 m	
South wall (layer 1)	2 cm plaster panel	
South wall (layer 2)	12 cm insulation [0.04 W/m·K]	5 cm insulation [0.08 W/m·K]
South wall (layer 3)	3 cm wood	
Window area (incl.frames)	5.10 m^2 (frames area: 17%)	
Glazing	SHGC=0.4; U-value=1.1 W/m^2K (same U-value for the frames)	
Blinds	Textile; 20% solar gains transmission when drawn	
Partition walls (internal)	10 cm (concrete bricks)	5 cm (concrete bricks)
Floor screed	6 cm (concrete)	5 cm (concrete)
Ceiling slab	25 cm (concrete)	15 cm (concrete)

Table 5.7: Main characteristics of the two different rooms considered in the simulations.

5.6 Energy Saving Potential

We test the heating control system making the hypothesis that the building is located in a central European city over a two-month period during the heating season (January to early March). We run simulations using the physics of the building theory in order to explore the energy saving potential of the Activity prediction. This work has been realized in collaboration with the LESO-PB laboratory of EPFL using an internally developed simulation framework [62].

5.6.1 Simulation Framework

Simulations are based on a basic dynamic nodal model which takes into account the heat capacity of the construction elements and of the room air. In our case, we use a basic dynamic nodal model having 13 nodes, as explained in [62]. The nodes represent a specific element in the environment: the first node represents the air in the room, the central nodes represent every wall and the last node represents the air outside the room. The simulation consists in computing the heat exchange taking place between all the nodes. Other factors, as the heat gains due to the sun and the activities are also considered at every step of the simulation.

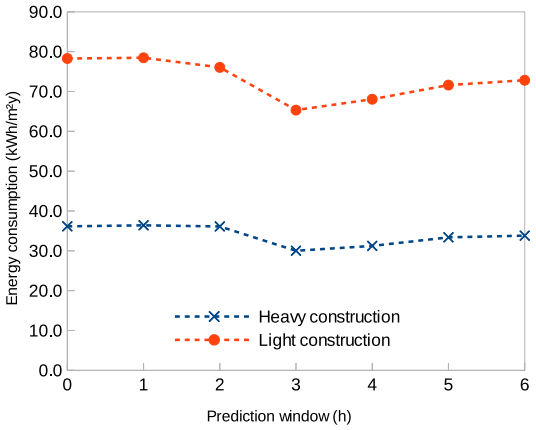
We simulate two different building types having a different thermal mass: a heavy construction with a good insulation and a lighter construction with a poorer insulation. In Table 5.7 we illustrate the main characteristics of the building. In the simulation we include the meteorological data synthetically generated by the Meteonorm software [63]. Data consists on hour of the day, month of the year, external temperature, global horizontal and global vertical south solar radiation for the months of January, February and early March (for a total of 65 days) matching the meteorological conditions of the city of Lausanne, Switzerland.

The heating controller consists in a closed-loop system based on the internal air temperature. We assign different set points for the temperature depending on the activity: (i) 20°C for normal home activities *(Activity 3)*, (ii) 17°C for night setback *(Activity 2)* and (iii) 12°C when absent *(Activity 1)*. During the simulation, the controller checks if the temperature is below the set point and below the setpoint of the future activity computed through the prediction window (1-6 hours). If both conditions are verified, then the necessary amount of energy in the room is injected so that the temperature can reach the set point value of the predicted activity.

For the blinds control, a simple closed-loop on/off control is used based on internal air temperature T_{int}. If the T_{int} is lower than 25°C, blinds are completely open for increasing the solar gains. In the opposite case, when the temperature is over this threshold, blinds are completely drawn for reducing the solar gains.

Prediction window	Heavy construction: Energy consumption [kWh/m^{2}y]	Heavy construction: Energy savings	Light construction: Energy consumption [kWh/m^{2}y]	Light construction: Energy savings
0h	36.1	-	78.3	-
1h	36.4	-0.70%	78.5	-0.23%
2h	36.1	0.10%	76.1	2.84%
3h	30.0	16.98%	65.3	16.55%
4h	31.2	13.60%	68.1	13.07%
5h	33.4	7.65%	71.6	8.53%
6h	33.8	6.45%	72.8	6.97%

(a)



(b)

Figure 5.23: Energy consumption and savings achieved for two different building construction types (a) when varying the prediction windows of activities (b).

5.6.2 Results

We estimate the energy consumption for the two buildings when using 6 different prediction time windows (1-6 hours). The results are illustrated in Figure 5.23. Results are compared to the energy consumption of the same buildings without activity prediction, i.e. when the prediction window is equal to 0.

As expected, the prior knowledge (prediction) of the activities has a positive effect on the energy consumption in buildings. Energy savings of up to 17% can be achieved in different types of buildings for a prediction window of 3 hours. However, as shown in Figure 5.23, prediction times over 3 hours result in lower energy savings. Although this may seem at first counter-intuitive from a building physics point of view (where we know that longer prediction times result in higher energy savings [64]), it can be justified when looking at the confusion matrices of the different prediction windows (Table 5.8). The recognition rate of *Activity 1* is rather moderate for prediction times of 1 to 3 hours (60-64%) while it drops further as the prediction window becomes bigger (50-57% for 4 to 6 hours). In addition, the confusion matrices show that incorrect predictions of this activity are almost all attributed to *Activity 3*. Since *Activity 1* is coupled with a significantly lower energy demand for space heating (temperature set point of 12°C) than the wrong predicted *Activity 3*, we observe this impact of the

Prevision 1 hour	Activity 1	Activity 2	Activity 3
Activity 1	.64	.01	.35
Activity 2	0	.85	.15
Activity 3	.11	.11	.78

Prevision 2 hours	Activity 1	Activity 2	Activity 3
Activity 1	.6	0	.4
Activity 2	.01	.94	.05
Activity 3	.18	.19	.63

Prevision 3 hours	Activity 1	Activity 2	Activity 3
Activity 1	.62	.01	.37
Activity 2	.02	.93	.05
Activity 3	.18	.2	.62

Prevision 4 hours	Activity 1	Activity 2	Activity 3
Activity 1	.55	.01	.44
Activity 2	.01	.91	.08
Activity 3	.14	.16	.7

Prevision 5 hours	Activity 1	Activity 2	Activity 3
Activity 1	.57	.01	.42
Activity 2	.01	.93	.06
Activity 3	.16	.19	.65

Prevision 6 hours	Activity 1	Activity 2	Activity 3
Activity 1	.5	.16	.48
Activity 2	.01	.94	.05
Activity 3	.12	.22	.66

Table 5.8: Confusion Matrices for different hours of prevision, using the three classes of activities.

low recognition rate on the energy savings. In simpler terms, the heating controller tends to heat up the building more than needed (as the prediction window becomes larger) when actually the occupants are absent.

5.7 Conclusions

In this chapter, we presented a system able to perform Activity prediction for the application of energy saving measures. HMMs were used for segmenting activities inside the in-home environment. We used a labeled database containing data acquired from 31 motion sensors and 4 door openings. We performed an activity recognition by using a HMM able to model 8 different activities, namely *relaxing*, *eating*, *working*, *sleeping*, *no activity*, *toilet*, *out of home* and *meal preparation*. We fed the HMM with observation sequences lasting one day and, through the Viterbi algorithm, we obtained the sequence of activities performed.

One of the main limitations of standard HMMs in this context is that the activity duration is not taken into account. As a consequence, HMMs performances tend to degrade when having long activities. This problem is in a certain way related to the Markovian hypothesis, that limits the computation of the transition to the current time. We tried to relax the Markovian hypothesis by injecting the information about the activity duration into the model. In fact, we applied the minimum duration models, consisting in modifying the topology of the HMM after the training. Minimum duration models consist in repeating single states several times with the aim of forcing the algorithm to spend a minimum time in each state. The information about the length of the activities was computed through the quantile length approach.

Then we compared both approaches, i.e. standard HMMs and minimum duration models, in terms of accuracy rate. By using minimum duration modeling we gained about 2% of accuracy rate when compared to standard HMMs. We reported the confusion matrices and, when using minimum duration models, we noticed an improvement in the segmenting capability of the algorithm, i.e. the capability to correctly identify the *No Activity* categories. By using minimum duration modeling the number of small segments misclassified was limited and the system improved in detecting if an activity was ongoing or not.

As following step, we performed the prediction of future activities by using an ANN, trained to find the relation between past and future activities. The ANN takes as input two windows of different length and sampling frequency on the past samples. Every 30 minutes we tried to forecast the activity performed in 6 different future times, ranging from 1 to 6 hours with a step of 1 hour. We compared such predictions with other predictive methods directly applied on features, i.e. a HMM and ANN. The ANN applied on the labels computed by HMMs provided better results than HMMs on features and similar results to the ANN on features.

As last step, we used the predictions for simulating the energy consumption savings on two buildings with different thermal mass, i.e. a heavy construction with a good insulation and a lighter construction with a poorer insulation. When using a prediction of 3 hours, a 17% of energy saving was achieved. By analyzing the confusion matrices, we observed that the energy saving results were particularly related to the recognition of a specific class, i.e. *out of home*. In fact, heating controllers heat up the building more than needed when actually the occupants are absent.

By comparing our work with others in literature, we noticed that only few contributions evaluated the machine learning approaches in terms of energy savings. Usually researchers in the Appliance recognition field build very complex algorithms without providing details about the benefit of their modeling techniques. We demonstrated that a well-know machine learning technique as HMMs, successfully yielded energy savings when including the information about the duration of the activities. A limitation of our analysis is the fact that we should somehow evaluate the reduction of comfort when we wrongly predict an activity.

The generative property of HMMs could provide some benefits. The property of the scalability could be useful also in this case. A new activity, corresponding to a new state, could be separately trained and added to the existing topology. This aspect could be advantageous when dealing with large quantity of data. Another interesting property of generative modeling concerns the HMMs scores. We could use the scores of HMMs for retrieving the information about the quality of the observations compared to those in the training set. If the observation sequence has a very low likelihood, something

different from the usual patterns is happening. For instance, in case of elderly surveillance, this information could be used for detection of abnormalities.

Bibliography

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Building*, vol. 40, pp. 394–398, 2008.

[2] G. Singla, D. Cook, and M. Schmitter-Edgecombe, "Tracking activities in complex settings using smart environment technologies," *International Journal of Biosciences Psychiatry and Technology*, vol. 1, pp. 25–35, 2009.

[3] A. Fleury, M. Vacher, and N. Noury, "SVM–based multimodal classification of activities of daily living in Health Smart Homes: sensors, algorithms, and first experimental results," *IEEE transactions on information technology in biomedicine*, vol. 14, pp. 274–283, 2010.

[4] H. Medjahed, D. Istrate, J. Boudy, and B. Dorizzi, "Human activities of daily living recognition using fuzzy logic for elderly home monitoring," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 2001–2006, 2009.

[5] U. Wilke, *Probabilistic Bottom-up Modelling of Occupancy and Activities to Predict Electricity Demand in Residential Buildings*. PhD thesis, ENAC, Lausanne, 2013.

[6] M. Alam, M. Reaz, and M. Ali, "A review of Smart Homes: past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, pp. 1190–1203, Nov 2012.

[7] C. Yiu and Y. Yau, "A learning model of intelligent home," *Facilities*, vol. 24, no. 9, pp. 365–375, 2006.

[8] F. Aldrich, "Smart Homes: past, present and future," in *Inside the Smart Home* (R. Harper, ed.), pp. 17–39, Springer London, 2003.

[9] D. Ding, R. Cooper, P. Pasquina, and L. Fici-Pasquina, "Sensor technology for smart homes," *Maturitas*, vol. 69, no. 2, pp. 131 – 136, 2011.

[10] I. Bierhoff, A. V. Berlo, J. Abascal, B. Allen, A. Civit, K. Fellbaum, E. Kemppainen, N. Bitterman, D. Freitas, and K. Kristiansson, *Towards an inclusive future: Impact and wider potential of information and communication technologies*, ch. Smart home environment. COST, Brussels, 2007.

[11] M. Mozer, "The neural network house: an environment that's adapts to its inhabitants," *Proceedings of the AAAI Spring Symposium Intelligent Environments*, pp. 110–114, 1998.

[12] H. Hagras, V. Callaghan, M. Colley, and G. Clarke, "A hierarchical fuzzy genetic multi-agent architecture for intelligent buildings online learning , adaptation and control," *Information Sciences*, vol. 150, pp. 33–57, 2001.

[13] E. Mathews, C. Botha, D. Arndt, and A. Malan, "HVAC control strategies to enhance comfort and minimise energy usage," *Energy and Buildings*, vol. 33, no. 8, pp. 853–863, 2001.

[14] M. Krarti, P. Erickson, and T. Hillman, "A simplified method to estimate energy savings of artificial lighting use from daylighting," *Building and Environment*, vol. 40, no. 6, pp. 747–754, 2005.

[15] V. Garg and N. Bansal, "Smart occupancy sensors to reduce energy consumption," *Energy and Buildings*, vol. 32, no. 1, pp. 81–87, 2000.

[16] C. Balaras, A. Gaglia, E. Georgopoulou, S. Mirasgedis, Y. Sarafidis, and D. Lalas, "European residential buildings and empirical assessment of the hellenic building stock, energy consumption, emissions and potential energy savings," *Building and Environment*, vol. 42, no. 3, pp. 1298–1314, 2007.

[17] L. Chen and I. Khalil, "Activity recognition: Approaches, practices and trends," *Activity Recognition in Pervasive Intelligent Environments*, vol. 4, pp. 1–31, 2011.

[18] "Residential Energy Consumption Survey (RECS)." `http://www.eia.gov/consumption/residential/`, 2009.

[19] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: Using occupancy sensors to save energy in homes," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, (New York, NY, USA), pp. 211–224, ACM, 2010.

[20] R. Arora, "Enabling context-aware applications in assistive environments," Master's thesis, University of Texas at Arlington, 2009.

[21] P. Rashidi and D. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, pp. 949–959, Sept 2009.

[22] G. Singla, D. Cook, and M. Schmitter-Edgecombe, "Incorporating temporal reasoning into activity recognition for smart home residents," in *Proceedings of the AAAI Workshop on Spatial and Temporal Reasoning*, 2008.

[23] H. Fang, R. Srinivasan, and D. Cook, "Feature selections for human activity recognition in smart home environments," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 3525–3535, 2012.

[24] E. Nazerfard, B. Das, L. B. Holder, and D. J. Cook, "Conditional random fields for activity recognition in smart environments," in *Proceedings of the 1st ACM International Health Informatics Symposium*, IHI '10, pp. 282–286, ACM, 2010.

[25] D. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intelligent Systems*, vol. 27, pp. 32–38, Jan 2012.

[26] G. Singla and D. Cook, "Interleaved activity recognition for smart home residents," *Intelligent Environment*, pp. 145–152, 2009.

[27] T. V. Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 1–9, ACM, 2008.

[28] D. Monekosso and P. Remagnino, "Monitoring behavior with an array of sensors," *Computational Intelligence*, vol. 23, no. 4, pp. 420–438, 2007.

[29] D. Monekosso, P. Remagnino, and Y. Kuno, "Anomalous behavior detection: Supporting independent living," in *Intelligent Environments*, Advanced Information and Knowledge Processing, pp. 33–48, Springer London, 2009.

[30] B. Dong and B. Andrews, "Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings," in *Proceedings of the 11th International Building Performance Simulation Association (IBPSA)*, pp. 1444–1451, 2009.

[31] D. Hu and Q. Yang, "Cigar: Concurrent and interleaving goal and activity recognition," in *Proceedings of the 23rd National Conference on Artificial Intelligence*, AAAI'08, pp. 1363–1368, 2008.

[32] S.-L. Chua, S. Marsland, and H. Guesgen, "Spatio-temporal and context reasoning in smart homes," in *Proceedings of the International Conference on Spatial Information Theory*, pp. 9–20, 2009.

[33] G. Singla, D. Cook, and M. Schmitter-Edgecombe, "Recognizing independent and joint activities among multiple residents in smart environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, pp. 57–63, 2010.

[34] Y.-T. Chiang, J.-J. Hsu, C.-H. Lu, L.-C. Fu, and J.-J. Hsu, "Interaction models for multiple-resident activity recognition in a smart home," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3753–3758, Oct 2010.

[35] L. Wang, T. Gu, X. Tao, and J. Lu, "Sensor-based human activity recognition in a multi–user scenario," in *Proceedings of the European Conference on Ambient Intelligence (AmI)*, pp. 78–87, 2009.

[36] A. Crandall and D. Cook, "Using a hidden markov model for resident identification," in *Proceedings of the 6th International Conference on Intelligent Environments (IE)*, pp. 74–79, 2010.

[37] A. Crandall and D. Cook, "Behaviometrics for identifying smart home residents," in *Human Aspects in Ambient Intelligence*, vol. 8 of *Atlantis Ambient and Pervasive Intelligence*, pp. 55–71, Atlantis Press, 2013.

[38] V. Srinivasan, J. Stankovic, and K. Whitehouse, "Using height sensors for biometric identification in multi-resident homes," in *Proceedings of the 8th International Conference on Pervasive Computing*, pp. 337–354, Springer-Verlag, 2010.

[39] N. Nguyen, D. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden markov models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 955–960, 2005.

[40] C. Wojek, K. Nickel, and R. Stiefelhagen, "Activity recognition and room-level tracking in an office environment," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 25–30, 2006.

[41] X. Ma, D. Schonfeld, and A. Khokhar, "Distributed multi-dimensional hidden markov model: theory and application in multiple-object trajectory classification and recognition," in *In Conference of Multimedia Content Access: Algorithms and Systems II (SPIE)*, 2008.

[42] D. De, W.-Z. Song, M. Xu, C.-L. Wang, D. Cook, and X. Huo, "Findinghumo: Real-time tracking of motion trajectories from anonymous binary sensing in smart environments," in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 163–172, June 2012.

[43] N. Noury and T. Hadidi, "Computer simulation of the activity of the elderly person living independently in a health smart home," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 3, pp. 1216–1228, 2012.

[44] T. Duong, H. Bui, Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi–markov model," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 838–845, 2005.

[45] S. Zhang, S. McClean, B. Scotney, P. Chaurasia, and C. Nugent, "Using duration to learn activities of daily living in a smart home environment," in *Proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare*, 2010.

[46] S. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[47] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Human Behavior Recognition with Generic Exponential Family Duration Modeling in the Hidden Semi-Markov Model," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, pp. 1–6, 2006.

[48] T. Van Kasteren, *Activity Recognition for Health Monitoring Elderly using Temporal Probabilistic Models.* PhD thesis, Amsterdam University, FNWI: Informatics Institute, 2011.

[49] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, "Collecting and disseminating smart home sensor data in the casas projec," in *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009.

[50] S. Intille, K. Larson, E. Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live–in laboratory for ubiquitous computing research," in *Pervasive Computing*, vol. 3968 of *Lecture Notes in Computer Science*, pp. 349–365, Springer Berlin Heidelberg, 2006.

[51] E. Tapia, S. Intille, L. Lopez, and K. Larson, "The design of a portable kit of wireless sensors for naturalistic data collection," in *Pervasive Computing*, vol. 3968 of *Lecture Notes in Computer Science*, pp. 117–134, Springer Berlin Heidelberg, 2006.

[52] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *Proceedings of the 9th International Conference on Ubiquitous Computing*, pp. 483–500, Springer-Verlag, 2007.

[53] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive'10, pp. 283–300, 2010.

[54] D. Cook, "Learning setting–generalized activity models for smart spaces," *IEEE Intelligent Systems*, vol. 27, pp. 32–38, 2010.

[55] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, vol. 103. Prentice Hall, 1993.

[56] F. Slimane, R. Ingold, A. Alimi, and J. Hennebert, "Duration Models for Arabic Text Recognition Using Hidden Markov Models," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control & Automation*, pp. 838–843, 2008.

[57] F. Einsele, R. Ingold, and J. Hennebert, "A Language-Independent , Open-Vocabulary System Based on HMMs for Recognition of Ultra Low Resolution Words," in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing*, pp. 429–433, 2008.

[58] M. Zimmermann and H. Bunke, "Hidden Markov Model Length Optimization for Handwriting Recognition Systems," in *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, pp. 1–6, 2002.

[59] S. Das, D. Cook, A. Battacharya, E. Heierman, and L. Tze-Yun, "The role of prediction algorithms in the mavhome smart home architecture," *IEEE Wireless Communications*, vol. 9, no. 6, pp. 77–84, 2002.

[60] G. Zhang, E. Patuwo, and M. Y.H., "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[61] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, pp. 318–362, 1986.

[62] N. Zarkadis, *Novel models towards predictive control of advanced building systems and occupant comfort in buildings.* PhD thesis, ENAC, Lausanne, 2015.

[63] "Meteonorm official website." `http://meteonorm.com/home/`, 2015.

[64] N. Morel, M. Bauer, M. El-Khoury, and J. Krauss, "Neurobat, a predictive and adaptive heating control system using artificial neural networks," *International Journal of Solar Energy*, vol. 21, no. 2-3, pp. 161–201, 2001.

Chapter 6

Conclusions

In this thesis, we investigated the use of generative modeling for classification tasks in the context of in-home monitoring time-series. Three types of signal and applications were investigated: (i) the appliance recognition using electrical appliance consumptions, (ii) the detection of glaucoma from IOP-related data and (iii) the activity recognition and prediction for energy optimization while preserving human comfort. We analyzed mainly two generative approaches: Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). Both models are able to capture stochastic properties of time series, which also means that the distribution of the observation sequences are modeled through parametric functions, in our case mixture of Gaussians. While GMMs are stateless, HMMs naturally attempts to capture the eventual state-based nature of time series.

In the Introduction we presented some general and specific research questions. Here below we summarize our contributions in regards of these questions.

1. The first question concerned the advantages in using generative approaches, such as GMMs and HMMs, in the context of time series acquired in in-home monitoring applications.

   In the Chapter 3 on the Appliance recognition, we extensively described the performances of K-NN, GMMs and HMMs modeling electrical signatures. We noticed that generative models outperform the K-NN, a discriminative non parametric approach. This difference in performances is mainly due to the sensitivity of K-NN to noisy observations. In fact, most of the electrical signatures present power features close-to-zero, corresponding to stretches of time where appliances are not active or in stand-by. These stretches are usually independent of the appliance and they are very close together in the feature space, leading to noisy neighbors for the K-NN. This effect is probably enhanced when using the dynamic coefficients, that introduce many close-to-zero samples, due to the steadiness of some signatures. Operations that reduce the number of close-to-zero samples, as the thresholding or the feature selection, yield significant improvements to the K-NN. On the other hand, when using generative modeling as GMMs and HMMs, the zero power stretches are compensating by bringing similar score contributions across categories. A similar effect is observed for steady part of the signals and when using dynamic coefficients.

   Generative models compute the likelihood that an observation sequence belongs to a given category. An advantage of such models against other approaches is that the meaning of this score is easy to interpret even for persons not used to statistic or machine learning. This property was particularly important in the case of the glaucoma detection, where proposing an easy and intuitive score to the ophthalmologists could help the collaboration between data scientists and health-care personnel. This collaboration is also important in a field where computer science goes hand in hand with medical knowledge.

2. The second question was about the analysis of the natural properties of generative approaches for performing tasks otherwise impossible or hard to make by using other methods such as discriminative approaches. By nature, generative models are scalable, adaptable and distributable. They are scalable when increasing the number of classes. If we add a new class to an already trained system, only the model corresponding to the new class has to be trained. Generative

models are also adaptable when the prior distribution of classes changes, for instance when using the system in a different context. This property is related to the fact that priors can be modified on the fly respecting the changes in the class distribution and according to the law of Bayes. Generative models are naturally distributable when the computational load has to be shared among different resources. This property is given by the fact that class models are naturally separated and therefore can be easily shared on different cpu resources.

In the Chapter 3 on Appliance recognition, we showed several applications taking advantage of these properties. A first application consisted in observing the scalability/adaptability of our system, when reducing the number of classes from 15 to 2. In that case we simply combined the scores of a GMMs classifier by selecting all the permutation of the chosen classes. A second application was the multi-signal identification, consisting in the appliance recognition when more than one appliance signature is merged on the same signal. When using discriminative modeling, the only solution consists in fully re-training the system in order to be able to recognize multiple appliances. We proposed an alternative solution by taking advantage of the additive property of electrical features. This property allows to merge the Gaussians of single models, providing an instantaneous training time. A third advantage concerns the distribution of the computation resources. We developed the concept of *Virtual Sensor* and *Virtual Classes*. The *Virtual Sensor* is a non-physical sensor that appears as a real one within the network. The *Virtual Classes* are non-physical systems where the generative machine learning models are stored. Their aim is to compute the likelihood associated to an observation. The *Virtual Sensor* compares the likelihoods coming from the different *Virtual Classes* to perform the identification of the winning class by using the Bayes' law.

In the Chapter 4 about the Glaucoma detection, we observed the advantage of generative modeling to face the problem of unbalanced classes. In fact, discriminative approaches intrinsically include the prior distribution of the classes and different strategies are often used to compensate this problem, such as artificially balancing the training set. In our case, we could handle seamlessly different priors for training and test sets using generative approaches by simply modifying the priors on the fly.

In the Chapter 5 on Activity recognition, we also assessed the advantages of the scalability. Adding a new activity, that corresponds to a new state, do not necessitate to re-train all the system. The state could be separately trained and later added to the existing topology. The benefit of such method is particularly significant when dealing with large quantity of data.

3. The third question was about the usefulness of state-base approaches, as HMMs, when compared to stateless approaches, as GMMs. In this case we obtained different answers depending on the application. If the signals under investigation have an underlying state-based structure, then HMMs, as expected, performed better. In the opposite case, HMMs provided similar performances to GMMs, that can be considered as the stateless version of HMMs. While these statements seem obvious, comparing GMMs and HMMs performances is bringing insights on the nature of the signals under investigation.

   In the Chapter 3 about Appliance recognition, HMMs showed limited improvements or similar results obtained with GMMs, in the case when appliances were already seen by classifiers. On the opposite case, when creating generic models able to classify appliances never seen before by classifiers, the accuracy rate increased with HMMs, showing the importance of state-based properties across brands of the same appliance category. In the Chapter 4 about Glaucoma detection, HMMs with two states performed only slightly better than GMMs. More complex models that consider a higher number of states did not improve the mean AUC. A specific topology using the information on sleep/awake periods did not show an increase of the performances. State base modeling was instead fundamental in the Chapter 5 about the Activity recognition. In that case, HMMs were used for segmenting the activities. Even if this approach has been previously tackled by other researchers, we also shown the possibility to include in HMMs duration information.

4. The fourth question concerned the restrictiveness of the observation independence hypothesis, that limit the computation of the probability on the observation x_t to the probability distribution of the state at the time t instead of the future and past observation. In practice, this

hypothesis says that the observation points are independent to each others. In most of cases, this hypothesis is false and its fairness depends on the data under analysis. However, some alternative solutions have been developed for counterbalancing this limitation. In this thesis, we used the computation of dynamic coefficients. The dynamic coefficients, called *delta* or *velocity* coefficients and *delta-delta* or *acceleration* coefficients, give information about the evolution of the observation sequences. These coefficients are similar to the first and second derivative.

In the Chapter 3 about the Appliance recognition, we tested the effectiveness of dynamic coefficients by using K-NN and generative models. While for the K-NN the dynamic coefficients seemed to introduce some noise, when using generative modeling as GMMs and HMMs, the models are able to take advantage of such coefficients. When great variations are verified in the electrical signatures, typically corresponding to a change of state, the log-likelihood of the matching model slowly decreases, while it dramatically drops for the other un-matching models. In the Chapter 4 about Glaucoma detection, we evaluated the inclusion of dynamic coefficients. In this case, we noticed that the inclusion of velocity coefficients provided significant improvements of the AUC. On the opposite case, the inclusion of the acceleration coefficients did not seem to further improve the AUC.

5. The fifth question was about the restrictiveness of the Markovian hypothesis in HMMs. The Markovian hypothesis limits the computation of the probability on the observation x_t to the transition from the state time t to the state at time $t-1$. In practice, only the transitions between the current and the past states are considered by HMMs. By using the minimum duration models we argue that we can somehow relax the Markovian hypothesis. Minimum duration models consist in a modification of the HMMs topology. In practice, HMMs states are repeated a number of times proportional to the duration of the corresponding hidden variable. As a consequence, the Viterbi algorithm is "forced" to spend a minimum time interval in a state.

   In the Chapter 5 about Activity recognition, we applied minimum duration models and we compared the performances against standard HMMs. We observed a systematical gain when using the minimum duration modeling, even when using different training procedures. An advantage of this approach against, for example, the Semi Hidden Markov Models, is that minimum duration modeling has the property to be fast to train and easy to implement.

We now summarize further contributions to the questions more specific to the domain of applications analyzed in this thesis.

1. The first specific question was about the possibility to discover appliance signatures allowing their identification. This question concerns the appliance recognition in the Intrusive Load Monitoring domain, that we have systematically explored. At the beginning of our work, only few research teams had previously attempted to directly address such signals in a systematic way. Observing a lack of publicly available data, we created our own database and we gave free access to the scientific community. The database comes along with four protocols, created in order to compare the algorithms results. At the time of writing this thesis, the database has requested by more than 50 researchers across academic and industrial organizations. The recognition of appliances already seen by classifiers yielded a significant accuracy rate (around 97%) and it could potentially be used in a realistic scenario. However, when we analyzed the recognition of appliances never seen before by classifiers, the accuracy rate appeared fair enough considering the difficulty of the task, but probably still too low for a realistic scenario (around 76%).

2. The second specific question concerned the effectiveness of the discrimination between healthy and glaucomatous patients by using the IOP-related data. As best result we obtained an AUC around 0.74 that can be considered a fair enough results considering the difficulty of the task. One limitation lies in the set of features. We focused on features that could be easily interpreted by the ophthalmologists, however other more complex features could potentially be included for improving the AUC. In addition, it should be considered that the ground truth corresponds to the decision of the ophthalmologists, deciding the condition of the patients' eye. Even if this is the best approach in order to determine the ground truth, errors of the ophthalmologists can not be excluded a priori. Regarding the modeling strategy, we tried to use a specific HMM

topology for observing if the patient state, i.e. “sleep” and “awake”, had an influence on the identification of glaucoma. We did not observe a direct improvement in terms of AUC when using this information. However, when using a linear alignment with two states, we observed some improvements. Other more pertinent separations may exist and this aspect could be analyzed deeper in future works.

3. The last specific question was about the meaningfulness of using a minimum duration modeling in the activity recognition context. As previously said, the minimum duration modeling helps in relaxing the restrictiveness of the Markovian hypothesis. The application of such modeling technique had a direct impact on the activity recognition accuracy rate (about 2-3%) and, as a consequence, on the prediction accuracy rate and on the energy saving then obtained.

   By comparing our work on the Activity recognition with others in literature, we noticed that only few contributions provided an evaluation of the machine learning approaches in terms of energy savings. In the literature, it is quite common to find papers in this field using very complex machine learning techniques without evaluating the pertinence in terms of energy reduction. In our case, we analyzed and proposed an approach to evaluate the energy gain through a building physics simulator. By using the minimum duration modeling approach, we were able to provide a gain of 17% in terms of energy reduction. A limitation of our analysis is the lack of evaluation of the comfort reduction when an activity is wrongly predicted.

   More detailed conclusions are available at the end of the Chapter 3 (Appliance Recognition), Chapter 4 (Glaucoma Detection) and Chapter 5 (Appliance Recognition for Energy Savings).

Appendix A

GMMs/HMMs Library

A.1 Introduction

In this appendix we present our contribution in terms of code. During this thesis, several efforts went to the implementation of the code. In practice, we wrote more than 22000 lines of code, for the most part dedicated to the implementation of machine learning algorithms. The principle was that, for really understanding an algorithm, you should be able to implement it from scratch. This operation was obviously time consuming, however it was very helpful in order to completely understand the steps of an algorithm and to have a full control on all its parameters. More specifically, in Table A.1, we report the algorithms used during this thesis. The large majority of the code has been implemented and/or used in Matlab (version 2014a/2015a). In specific, we used the pre-existing Toolboxes of Matlab for the ANNs and the Decision Trees. The code for the GMMs/HMMs and for the K-NN was implemented from scratch. In particular, some functions were implemented in C and included in Matlab for speeding up the computational time. We completed these C functions for realizing a stand alone version of our implementation that could work on the OpenPicus and Rasperry Pi for the testing phase, as presented in Chapter 3. In addition, we realized also some code implementations in different languages, as Python, Java and C++, mostly to perform specific pre-processing tasks depending on the source of data.

Algorithm	**Language**	**Toolbox**
K-NN	Matlab	—
Decision Tree	Matlab	Statistics Toolbox
ANN	Matlab	Neural Network Toolbox
GMMs/HMMs	Matlab	—
GMMs/HMMs (Test)	C	—

Table A.1: Information about the machine learning algorithms used during this thesis and the origin of the algorithms. The GMMs/HMMs and K-NN have been implemented from scratch.

The majority of our efforts went in the realization of the GMMs/HMMs library in Matlab. We are planning to give free access to our code, as soon as the release version will be ready. In this appendix we provide more details on this library. We start by introducing the glossary of terms and continue by describing specific data structures used by the library. The main functions are **HMMTraining** and **HMMTesting**, dealing respectively with the training of models, through the computation of the transition and emission matrices, and the classification of the observation sequences by choosing the most probable model. Other functions, as **CheckData**, **Reduction** and **Normalization**, will be shortly presented.

A.2 Glossary and data structures

Hereafter we report a small glossary useful when reading this appendix:

- M is the total number of models and m a generic model;
- DTr is the total number of observations in the training set and i is a generic observation sequence in the training set.

- DTs is the total number of observations in the test set and j is a generic observation sequence in the test set.
- P is the dimension of the feature space;
- T is the total number of samples in a specific observation sequence, i.e. the length of the observation sequence;
- K is the total number of Gaussians;
- $Mode$ is type of topology, it can be forward single-step ($Mode = 0$), ergodic ($Mode = 1$) or personalized ($Mode = 2$);
- N is the total number of states and n is a generic state. There are always the start and the end state, i.e. the non-emitting states, therefore $N > 2$.

Four main data structures are listed below:

- **cTr**, the training set. It is a vector containing DTr observation sequences. Each observation sequence $cTr\{i\}$ has three fields:
  - *.c*, is a matrix of dimension $P \times T$, containing the observation sequence. While the total number of samples in a specific observation sequence can differ between different observation sequences, the number of dimensions has to be the same.
  - *.name* (optional), is the name of the observation sequence.
  - *.label* (optional), is the labeling of the states. It can be used for deciding the initial separation of the observations vectors to the states.
- **cTs**, the test set. It is a vector of DTs observation sequences. Each observation sequence $cTs\{j\}$ has the same three fields of $cTr\{i\}$:
  - *.c*, is a matrix of dimension $P \times T$, containing the observation sequence. While the total number of samples in a specific observation sequence can differ between different observation sequences, the number of dimensions has to be the same.
  - *.name* (optional), is the name of the observation sequence.
  - *.label* (optional), is the labeling of the states. It can be used for evaluating the performances of the Viterbi alignment.
- **labelTr** is a vector of length $1 \times DTr$ containing the labels of the models. Labels vary between 1 and M.
- **labelTs** is a vector of length $1 \times DTs$ containing the labels of the models. Labels vary between 1 and M.

A.3 Main functions

The main functions are: (i) Configuration, (ii) HMMTraining, (iii) HMMTesting. Other support functions have been also implemented.

A.3.1 Configuration

It reads an external file containing all the information that can be eventually adapted by the user. It contains several options, as shown in the example below.

```
% Basic parameters
N:[5 4 5 5]
K:[16 16 16 16]
Mode:[1 1 1 1]
varFloor:1e-4

% Reduction
Red:[2 4]

% Normalization
norm:1

% Dynamic coefficients
Wind:2

% K-means
Kmeans_replicate:200
Kmeans_maxIter:10

% Alignment
Align:0

% Expectation Maximization
EM_maxIter:100
EM_conv:1e-5
EM_relevFactor:16

% Hidden Markov Model
HMM_maxIter:50
HMM_conv:1e-5
Info_disp:1
Plot_prob:0

% Results
Print_results:1
Plot_resViterbi:0
```

All this information is loaded in a structure of data, called *Opt*. For instance, if we want access to the normalization value, we can recover this information by using $Opt.norm$. The information on the number of states $Opt.N$, number of Gaussians $Opt.K$ and topologies $Opt.Mode$ are structured as vectors of dimension $1 \times M$, given that they can differ among the models.

A.3.2 HMMTraining

The observation sequences are divided in groups depending on the belonging model. This information is retrieved from the vector **labelTr**. The training of the M models is divided by three phases:

- Initialization of the transition matrix
- Initialization of the emission matrices
- Convergence of the model

Initialization of the transition matrix

We initialize the transition matrix of a model m by knowing the number of states and the topology, i.e. $Opt.N(m)$ and $Opt.Mode(m)$. As previously said, this number includes the non-emitting states, therefore $Opt.N(m) > 2$. We use discrete numerical values for describing the topologies:

$Opt.Mode(m) = 0$ is the forward, $Opt.Mode(m) = 1$ is the ergodic and $Opt.Mode(m) = 2$ is a personalized topology. We do not provide here more details on the personalized topology, however additional information on the topology would be required. The transition matrix, called **A**, is then generated. Hereafter we show two example of initialization of an ergodic ($Opt.Mode(m) = 1$) and a forward ($Opt.Mode(m) = 0$) matrices when $Opt.N(m) = 5$.

	start	**1**	**2**	**3**	**end**
start	0	0.33	0.33	0.33	0
1	0	0.25	0.25	0.25	0.25
2	0	0.25	0.25	0.25	0.25
3	0	0.25	0.25	0.25	0.25
end	0	0	0	0	0

Table A.2: Initialization when $Opt.Mode(m) = 1$ and $Opt.N(m) = 5$.

	start	**1**	**2**	**3**	**end**
start	0	1	0	0	0
1	0	0.5	0.5	0	0
2	0	0	0.5	0.5	0
3	0	0	0	0.5	0.5
end	0	0	0	0	0

Table A.3: Forward initialization when $Opt.Mode(m) = 0$ and $Opt.N(m) = 5$.

For the ergodic example, all transitions are allowed, with the exception of the transitions going in the start state, the transitions outgoing from the end state, and the direct transition start-end state. The weights are equally distributed among the outgoing transitions of each state. In Table A.2, we use 3 states above the start and end state ($Opt.N(m) = 5$).

For the forward example, allowed transitions are only the self-transitions and the ones to the next state. In Table A.3, we use 3 states in addition to the start and end state ($Opt.N(m) = 5$). At the beginning, there is the 100% of probability of going into the next state. For the other states, only one step forward or remaining in the same state is admitted, according to the left-right concept.

Initialization of the emission matrices

As next step, we initialize the emission matrices. The probability distributions are multivariate mixture of Gaussians, therefore, for every state, we have a matrix of means, a matrix of standard deviations and a vector of weights. We have a matrix of standard deviation given that we use a diagonal matrix, supposing the independence of the features. More specifically:

1. **MI** is a vector of the matrices of means of length N. Each element $MI\{n\}$ of the vector corresponds to the matrix of the means relative to a state. The matrix has dimension $P \times K$, where P is the feature dimension and K is the number of Gaussians relative to the n-th state.

2. **SIGMA** is a vector of the matrices of standard deviations. Each element $SIGMA\{n\}$ of the vector corresponds to the matrix of the standard deviations relative to a state. The matrix has dimension $P \times K$, where P is the feature dimension and K is the number of Gaussians relative to the n-th state.

3. **PComp** is a vector of the vectors of weights. Each element $PComp\{n\}$ corresponds to the vector of the weight relative to a state. It is independent of the dimension P.

In order to initialize the emission matrices, we use several information taken from the *Opt* file. In particular for each model m we use the number of states $Opt.N(m)$ and the number of Gaussians $Opt.K(m)$. The variance floor *Opt.varFloor* is a single variable that is equal across all the models.

If an observation $cTr\{i\}$ has the field *.label*, it takes this field for retrieving the relation between observations and states, in the other case a linear alignment is performed. With the options $Opt.align = 1$ we can effectuate a linear alignment if the field *.label* is present. A synthetic example is shown in Figure A.1. For each state, the observations having the corresponding label are grouped together. For each group, we compute a K-Means distribution, using as number of cluster the number of Gaussians $Opt.K(m)$. The K-Means separates the observations in clusters and we use this information for initializing the Expectation-Maximization (EM) algorithm. We repeat the K-Means a given number of times *Opt.Kmeans_replicate* and we choose the best solution. In order to speed up the procedure, we limit the numer of iterations of the K-Means to the value *Opt.Kmeans_maxIter*. The EM algorithm computes the most probable positions of $Opt.K(m)$ Gaussians, giving their means,

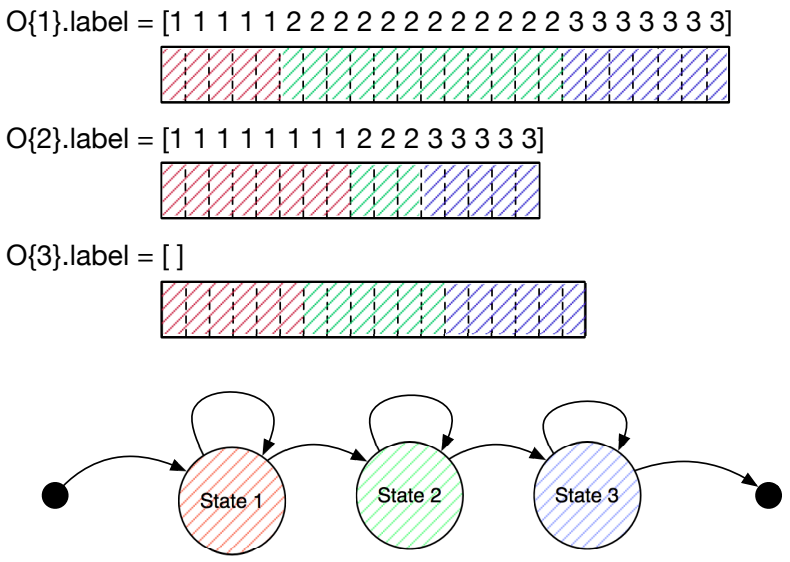


Figure A.1: Example of an alignment using the *.label* field for the two first observation sequences and a linear alignment in the third case.

standard deviations and weights. For the n-th state we store inside $MI\{n\}$ the matrix of means, inside $SIGMA\{n\}$ the matrix of standard deviations and inside $PComp\{n\}$ the vector of weights.

There are some parameters of the EM algorithm that are customizable:

- The algorithm converges under the threshold *Opt.EM_conv*.
- The number of iterations is greater than the predefined value *Opt.EM_maxIter*.

Estimation of new parameters

The set of parameters (**A**, **SIGMA**, **MI**, **PComp**) is iteratively updated by using the Viterbi algorithm for the assignation of the labels and the EM for the re-computation of the parameters. Similarly to the EM algorithm, this part is divided in two steps:

- **Assignation**. The assignation step is performed by the Viterbi algorithm. The probability calculation and the alignment are obtained using the function *viterbi_log_fast*, implemented in C for a faster execution. We compute the most probable sequence of states and its log-likelihood. For every observation sequence $cTr\{i\}$, we apply this function and we calculate the most probable sequence of states and its probability.
- **Re-computation**. We can now re-compute the HMM parameters. For the transition matrix, we use the Viterbi criterion. We compute all the transitions between the states after the Viterbi algorithm. By a simple normalization, we re-estimate the transition matrix. For the emission matrices, we re-compute the **SIGMA**, **MI** and **PComp**, using the EM algorithm. This time we simply use as label the alignment determined by the Viterbi algorithm.

Similarly to the EM, we stop the computation if:

- The model converges under the threshold *Opt.HMM_conv*. We use the sum of the probabilities coming from the Viterbi algorithm.
- The number of iterations is greater than the predefined value *Opt.HMM_maxIter*.

The *Opt.Plot_prob* plot the evolution of the probability during the training. The *Opt.Info_disp* simply plot additional information concerning the steps of the training.

A.3.3 HMMTesting

The testing of the observation sequences is relatively simple when compared to the training. We use the same function *viterbi_log_fast* used for the training on each observation sequence in the test set $cTs\{j\}$. The Viterbi algorithm computes the alignment and the relative probability. We take into account the probability in order to find the most probable model. For each observation sequence $cTs\{j\}$ we create a vector of probabilities, i.e. one per model, and we choose the most probable one as winner. We test every observation and we create a vector containing the list of winning models **labelComp**. By comparing this vector with the ground truth, i.e. **labelTs**, we are able to compute the confusion matrix. Given that the test data is not imperatively balanced among the classes, the confusion matrix is given by default using the number of observations and not the percentage.

The option *Opt.Print_results* prints each observation sequence with a different color depending if it has been correctly classified or not. The option *Opt.Plot_resViterbi* is able to plot the evolution of the log-likelihood for the observation sequence $cTs\{j\}$.

A.3.4 Support functions

Checkdata

This function checks the correctness of data structures and the coherence between data structures, labels and options. It also checks the coherency of the state vector $Opt.N$, topology vector $Opt.Mode$, and Gaussian vector $Opt.K$. In addition, all the feature dimensions of the observation sequences in the training and test sets must have the same dimension. The library is not able to manage the NaN values, essentially because of the EM algorithm, therefore these values are automatically removed. When a condition is not satisfied, the process stops and the explication is displayed in the terminal.

Reduction

This function permits to remove the feature(s) from the training and test sets. It is possible to indicate the feature(s) that can be eliminated, e.g. if the vector of dimension p is equal to [1 2 3 4 5] and *Opt.Red* is equal to [2 4], then, after the Reduction, p = [1 3 5]. This function is particularly useful for the feature selection.

Normalization

This function normalizes the data. The data are normalized using the information coming from the training set. Four different normalizations are available: z-score normalization using the means and variances of all the training data ($Opt.norm = 1$), min-max normalization using the min and max of all the training data ($Opt.norm = 2$), z-score normalization using the means and variances of the single observations ($Opt.norm = 3$) and min-max normalization using the min and max of the single observations ($Opt.norm = 4$).

DeltaCoeff and Delta2Coeff

These functions compute respectively the delta and delta-delta coefficients. The information about the length of the window used for the computation has to be specified in $Opt.Wind$, in terms of number of samples. The training and test sets are modified for expanding the feature space.

HMMMapping

This function is related to the MAP adaptation using a UBM model. Two training are possible: (i) use all data for training the UBM and (ii) use subpopulation of data for training distinct UBMs and then merge the models together. While the first method is conceptually easier, it usually requires a balanced data among the subpopulations. The second method on the contrary is robust to the unbalanced subpopulation, however it requires to merge the models. If the *Opt.EM_relevFactor* is > 0, then an UBM is computed.

HMMExpanded

This function has been made for modifying the HMM topology by repeating its states. In particular, three operations are possible: (i) minimum duration models, (ii) maximum duration models and (iii) Ferguson models. The information about state repetitions, in the form of a list, is sent to this function in the form of vector. As output, the parameters **A**, **SIGMA**, **MI** and**PComp** are modified accordingly with the list of state repetitions *Opt.Chain*, if present.

Other

Other functions are provided for performing some standard operations accordingly to our data structures. We implement functions for preforming the PCA, adding noise, computing the ROC curve and the AUC, etc.

Appendix B

Physics of the building

In this thesis, we focus on the machine learning tasks for providing energy saving measures. However, from another point of view, interventions on the building structure, e.g. the insulation of external walls or weather proofing of openings, can potentially reduce the energy consumption. However, each building has unique peculiarities that make it different from the others. The term *energy signature of buildings* indicates the specific characteristics of the building from an energetic point of view. In addition it provides important information for the potential energetic and cost-effective improvements.

The energy signature is built by observing the response of the building, in terms of energy consumption, to the external conditions. In Figure B.1 we illustrate the theoretical response of the building to the external temperature. Some sources of consumption are always present, independently on the external temperature, i.e. the water heating, lights and appliances. Other sources of consumption are present only when the outside temperature is under a certain threshold, i.e. ventilation, air infiltration and duct loss, ceiling heat loss, wall heat loss and window heat loss. Their influence is proportional to the external temperature. When the temperature is over a certain threshold, then the air conditioning is activated, representing another source of consumption.

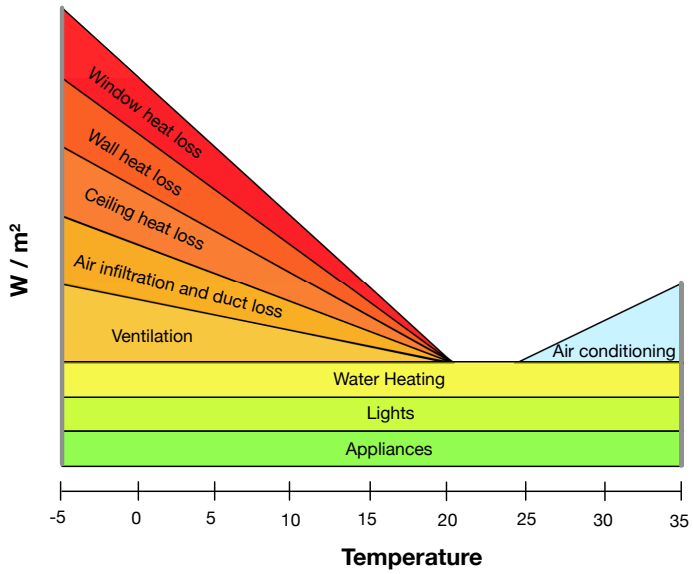
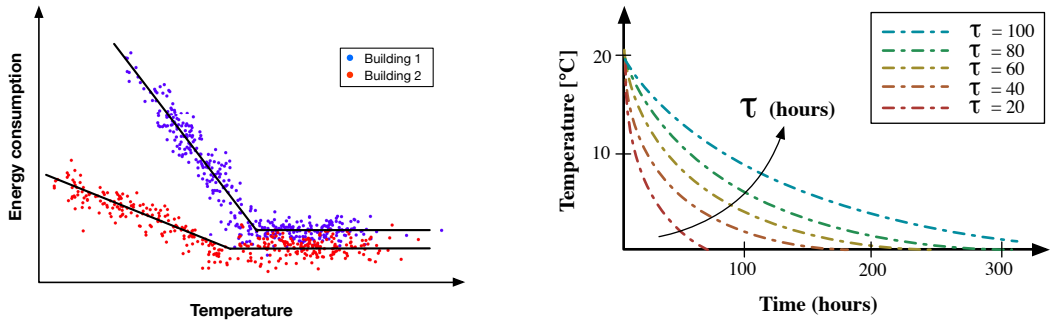


Figure B.1: Expected response of the building to the external temperature [1]. Details about the single contributions are provided.

Computing the building energy signature by using only the information on the building structural elements, as the insulation of the walls, the windows, etc. is a difficult task. It requires several detailed information and a model representing the structure of the building. A simpler solution is provided by a data-driven approach. The energy consumption of the building and the external temperature are measured on regular basis. The energy signature is therefore recovered by computing the curve that best fits the measures. The slope of the curve is particularly interesting, given that it represents the response of the building from the energetic point of view during the cold season. In Figure B.2a we illustrate an example of two buildings having a different slope. For a smaller slope, a smaller quantity



(a) Two buildings with different slopes are compared: the blue has an higher slope than the red one.

(b) Different behaviors of five buildings: the smaller the slope of the curve, the higher the τ.

Figure B.2: Two different types of analysis are presented: (a) the trend of the energy consumption when varying the external temperature and (b) the trend of the temperature when the heating is turned off at time zero with an external temperature of 0 °C and an internal temperature is 20 °C [2].

of energy is needed for heating up the building.

The thermal inertia of the building, described by the thermal time constant τ, can now be computed. This parameter describes how much time it takes to change the temperature inside a building if the heating system is suddenly interrupted. The thermal time constant is the ratio between the thermal mass of the building and the overall heat loss coefficients. The temperature inside the building can be represented by the following Formula:

$$\Delta T_t = \Delta T_0 \cdot (1 - exp^{-t/\tau}) \tag{B.1}$$

where ΔT_0 is the difference of temperature at time zero and ΔT_t is the temperature inside the building after the interruption of the heating system. In Figure B.2b we illustrate an example of five buildings with different τ. The curves are obtained by interrupting the heating system at time zero with an external temperature of 0 °C and an internal temperature of 20 °C.

In conclusion, each building has its own characteristics influencing the possible energy saving measures. For instance if the slope of the *energy signature of buildings* is steep, the building usually needs some structural interventions for reducing the slope. However, when the slope is feeble, structural interventions can be avoided for the benefit of automatic system for the energy management. In the latter case, predictive models can be used for the optimization of the energy consumption. According to the models above, we can infer that the greater the τ, the more effective the predictive models.

Bibliography

[1] J. White, "All electric home energy usage." http://homeenergypros.lbl.gov/group/buildingsimulations/forum/topics/using-average-monthly?xg_source=activity, 2012.

[2] J. Karlsson, "Possibilities of using thermal mass in buildings to save energy, cut power consumption peaks and increase the thermal comfort," Master's thesis, Lund Institute of Technology, Division of Building Materials, Sweden, 2012.

List of Figures

List of Tables

List of publications

[1] A. Ridi, C. Gisler, and J. Hennebert, "Automatic identification of electrical appliances using smart plugs," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)*, pp. 301–305, 2013.

[2] C. Gisler, A. Ridi, D. Zufferey, O. A. Khaled, and J. Hennebert, "Appliance consumption signature database and recognition test protocols," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)*, pp. 336–341, 2013.

[3] A. Ridi, C. Gisler, and J. Hennebert, "Unseen appliances identification," in *Proceedings of the 18th Iberoamerican Congress on Pattern Recognition (CIARP)*, pp. 75–82, 2013.

[4] A. Ridi, C. Gisler, and J. Hennebert, "A survey on intrusive load monitoring for appliance recognition," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[5] A. Ridi, C. Gisler, and J. Hennebert, "ACS-F2 - A New Database of Appliance Consumption Analysis," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SOCPAR)*, 2014.

[6] A. Ridi and J. Hennebert, "Hidden Markov Models for ILM appliance identification," in *Proceedings of the International Workshop on Enabling ICT for Smart Buildings (ICT-SB)*, 2014.

[7] A. Ridi, C. Gisler, and J. Hennebert, "Appliance and State Recognition using Hidden Markov Models," in *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, 2014.

[8] A. Ridi, C. Gisler, and J. Hennebert, "Processing smart plug signals using machine learning," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 75–80, March 2015.

[9] A. Ridi, C. Gisler, and J. Hennebert, "User interaction event detection in the context of appliance monitoring," in *Proceedings on the IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM Workshops)*, pp. 323–328, March 2015.

[10] G. Bovet, A. Ridi, and J. Hennebert, *Toward Web Enhanced Building Automation Systems - Big Data and Internet of Things: A Roadmap for Smart Environments*, vol. 546 of *Studies in Computational Intelligence*, ch. 11, pp. 259–284. Springer, 2014.

[11] G. Bovet, A. Ridi, and J. Hennebert, "Virtual things for machine learning applications," in *Proceedings of the 5th International Workshop on Web of Things (WOT)*, WoT, pp. 4–9, ACM, 2014.

[12] G. Bovet, A. Ridi, and J. Hennebert, "Appliance recognition on internet-of-things devices," in *Proceedings of the 4th International Conference on Internet of Things (WOT), Demonstration Session*, IoT, 2014.

[13] C. Gisler, A. Ridi, J. Hennebert, R. Weinreb, and K. Mansouri, "Automated detection and quantification of circadian eye blinks using a contact lens sensor," *Translation Vision Science and Technology*, vol. 4, no. 1, pp. 1512–1514, 2015.

[14] C. Gisler, A. Ridi, M. Fauquex, D. Genoud, and J. Hennebert, "Towards glaucoma detection using intraocular pressure monitoring," in *Proceedings of the 6th International Conference on Soft Computing and Pattern Recognition (SOCPAR)*, pp. 255–260, Aug 2014.

[15] M. Caon, F. Carrino, A. Ridi, Y. Yue, O. Khaled, and E. Mugellini, "Kinesiologic electromyography for activity recognition," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA)*, pp. 34:1–34:7, ACM, 2013.

[16] F. Carrino, A. Ridi, E. Mugellini, O. Khaled, and R. Ingold, "Gesture segmentation and recognition with an emg-based intimate approach - an accuracy and usability study," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on*, pp. 544–551, July 2012.

[17] F. Carrino, A. Ridi, R. Ingold, O. Abou Khaled, and E. Mugellini, "Gesture vs. gesticulation: A test protocol," in *Proceedings of the 15th International Conference on Human-Computer Interaction (HCI)*, pp. 157–166, 2013.

[18] F. Carrino, A. Ridi, M. Caon, O. Abou Khaled, and E. Mugellini, "Optimization of an electromyography-based activity recognition system," in *Proceedings of the IEEE Conference on Biosignals and Biorobotics Conference (BRC)*, pp. 1–6, Feb 2013.

[19] A. Ridi, N. Zarkadis, G. Bovet, N. Morel, and J. Hennebert, "Towards reliable stochastic data-driven models applied to the energy saving in buildings," in *Proceedings of the International Conference on Cleantech for Smart Cities and Buildings (CISBAT)*, 2013.

[20] N. Zarkadis, A. Ridi, and N. Morel, "A multi-sensor office-building database for experimental validation and advanced control algorithm development," in *The 4th International Conference on Sustainable Energy Information Technology (SEIT)*, pp. 1003 – 1009, 2014.

[21] A. Ridi, N. Zarkadis, C. Gisler, and J. Hennebert, "Duration models for activity recognition and prediction in buildings using hidden markov models," in *Proceedings on the 2nd IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, Oct 2015.