Complex event recognition through wearable sensors

THÈSE N^{O} 7653 (2017)

PRÉSENTÉE LE 12 MAI 2017
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES D'INFORMATION RÉPARTIS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Jean-Eudes Marie RANVIER

acceptée sur proposition du jury:

Prof. S. Süsstrunk, présidente du jury
Prof. K. Aberer, directeur de thèse
Prof. M. Schumacher, rapporteur
Dr S. Mukherjea, rapporteur
Prof. D. Gatica-Perez, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2017

Everything is a matter of reversibility
— Anonymous

To my siblings

Acknowledgements

First and foremost, i would like to thank my supervisor, Prof. Karl Aberer for giving me the opportunity to do a PhD in the LSIR laboratory. I am thankful for the autonomy he gave me while providing key supervision when needed.

I also would like to thank the committee members, Prof. Daniel Gatica-Perez, Prof. Michael Schumacher and Dr. Sougata Mukherjea for taking the time to review my thesis, and Prof. Sabine Süsstrunk for accepting to chair the committee.

I also would like to thank the people with whom I directly collaborated with: From EPFL, I thank Dr. Michele Catasta who was always here in time of need but not only, Dr. Matteo Vasirani, Dr. Zhixian Yan, Dr. Jean-Paul Calbimonte and Burak Zeydan. Outside of EPFL, I thank Fabien Dubosson and Stefano Bromuri from HES-SO Wallis, for the fruitful collaboration.

I also would like to thank the nurses from the hospital Riviera Chablais who were instrumental in acquiring data for the D1namo project, as well as the association vaudoise du diabète who relayed our call for study participants. I would like to extend my thanks to all the members of LSIR, for the interesting and fun discussions, with a special thank to Chantal Francois who helped me getting every piece of paperwork out of the way.

During this PhD I also met a lot of wonderful people in Lausanne, mostly other PhD students at EPFL. I want to thank them for all the fun they brought to Lausanne. I would like to specially thank Dr. Fay Prifti for supporting and motivating me every single day of this journey.

Last but not least I would like to express all my gratitude to my family for believing in me and motivating me for the past 28 years.

Lausanne, 17/02/2017 J-E.R.

Abstract

Complex events are instrumental in understanding advanced behaviours and properties of a system. They can represent more meaningful events as compared to simple events. In this thesis we propose to use wearable sensor signals to detect complex events. These signals are pertaining to the user's state and therefore allow us to understand advanced characteristics about her. We propose a hierarchical approach to detect simple events from the wearable sensors data and then build complex events on top of them.

In order to address privacy concerns that rise from the use of sensitive signals, we propose to perform all the computation on device. While this ensures the privacy of the data, it poses the problem of having limited computational resources. This problem is tackled by introducing energy efficient approaches based on incremental algorithms.

A second challenge is the multiple levels of noise in the process. A first level of noise concerns the raw signals that are inherently imprecise (e.g. inaccuracy in GPS readings). A second level of noise, that we call *semantic noise*, is present among the simple events detected. Some of these simple events can disturb the detection of complex events effectively acting as noise.

We apply the hierarchical approach in two different contexts defining the two different parts of our thesis.

In the first part, we present a mobile system that builds a representation of the user's life. This system is based on the episodic memory model, which is responsible for the storage and recollection of past experiences. Following the hierarchical approach, the system processes raw signals to detect simple events such as places where the user stayed a certain amount of time to perform an activity, therefore building sequences of detected activities. These activities are in turn processed to detect complex events that we call *routines* and that represent recurrent patterns in the life of the user.

In the second part of this thesis, we focus on the detection of glycemic events for diabetes type-1 patients in a non-invasive manner. Diabetics are not able to properly regulate their glucose, leading to periods of high and low blood sugar. We leverage signals (Electrocardiogram (ECG), accelerometer, breathing rate) from a sport belt to infer such glycemic events. We propose a physiological model based on the variations of the ECG when the patient has low blood sugar, and an energy-based model that computes the current glucose level of the user based on her glucose intake, insulin intake and glucose consumption via physical activity.

For both contexts, we evaluate our systems in term of accuracy by assessing wether the detected routines are meaningful, and wether the glycemic events are correctly detected, and in term of mobile performance, which confirms the fitness of our approaches for mobile computation.

Keywords: wearable sensor, mobile computing, complex event, event recognition, incremental algorithm, episodic memory, routine detection, non-invasive glucose monitoring

Résumé

Les événements complexes jouent un rôle déterminant dans la compréhension des comportements et des propriétés avancés d'un système. Ils peuvent représenter des événements qui ont d'avantage de sens comparés aux événements simples. Dans cette thèse, nous proposons d'utiliser des capteurs portables afin de détecter des événements complexes. Les signaux fournis par ces capteurs sont liés à l'état de l'utilisateur et nous permettent donc d'étudier ses caractéristiques avancées. Nous proposons une approche hiérarchique qui commence par la détection d'événements simples à partir des données des capteurs portables. Nous construisons ensuite les événements complexes à partir de ces évènements simples.

Afin de répondre aux préoccupations en matière de protection de la vie privée qui découlent de l'utilisation de signaux sensibles, nous proposons d'effectuer tous les calculs directement sur l'appareil mobile. Bien que cela assure la confidentialité des données, cela restreint également les ressources disponibles. Ce problème est résolu en introduisant des approches écoénergétiques basées sur des algorithmes incrémentiels.

Un deuxième défi concerne les différents niveaux de bruit dans le processus. Un premier niveau de bruit concerne les signaux bruts qui sont intrinsèquement perturbés (par exemple, les signaux GPS contiennent des inexactitudes). Un deuxième niveau de bruit, que nous appelons *bruit sémantique*, est présent au niveau des événements simples. Certains de ces événements peuvent en effet perturber la détection d'événements complexes agissant ainsi comme du bruit.

Nous appliquons l'approche hiérarchique dans deux contextes différents définissant les deux parties de notre thèse.

Dans la première partie, nous présentons un système mobile qui construit une représentation de la vie de l'utilisateur. Ce système est basé sur le modèle de mémoire épisodique, qui est responsable du stockage et de la remémoration des expériences passées. Suivant l'approche hiérarchique, le système traite les signaux bruts pour détecter des événements simples tels que des lieux où l'utilisateur a passé un certain temps pour effectuer une activité, en créant ainsi des séquences d'activités. Ces activités sont à leur tour traitées pour détecter des événements complexes que nous appelons des *routines* et qui représentent des séquences d'évènements récurrents dans la vie de l'utilisateur.

Dans la deuxième partie de cette thèse, nous nous concentrons sur la détection de manière non invasive des événements glycémiques pour les patients diabétiques de type-1. Les personnes diabétiques ne sont pas en mesure de réguler correctement leur niveau de glucose, conduisant à des périodes de haute et basse glycémie. Nous utilisons des signaux de capteurs (Electrocardiogramme (ECG), accéléromètre, capteur de respiration) intégrés dans une ceinture de sport afin de déduire de tels événements glycémiques. Nous évaluons un modèle physiologique basé sur les variations de l'ECG lorsque le patient a une faible glycémie, et un modèle qui calcule le niveau de glucose actuel de l'utilisateur sur la base de sa consommation de glucose, son apport en insuline et son utilisation du glucose due à une activités physiques.

Dans les deux contextes, nous évaluons nos systèmes en termes de précision en évaluant si les routines détectées sont significatives et si les événements glycémiques sont correctement détectés, et en terme de performance mobile, afin de confirmer la validité de nos approches pour effectuer les calculs directement sur l'appareil mobile.

Mots clefs : Capteur portable, mobile computing, événement complexe, reconnaissance d'événement, algorithme incrémentiel, mémoire épisodique, détection de routine, évaluation du glucose de manière non-invasive

Contents

Acknowledgements **i**

Abstract (English/Français/Deutsch) **iii**

List of figures **xi**

List of tables **xiii**

1 Introduction **1**

1.1 Challenges 2

1.1.1 Multi-level noise 2

1.1.2 Privacy preservation 3

1.1.3 Constrained resources 4

1.2 Research directions 4

1.2.1 Complex routine detection 4

1.2.2 Blood glucose level inference 5

1.3 Contributions 6

1.4 Thesis structure 7

1.5 Publications & contributions of the author 7

I Sensing user memories **11**

2 Background **13**

2.1 Introduction 13

2.2 Global vision 14

2.2.1 Episodic Memory Model 14

2.2.2 System architecture 16

2.3 State of the art 19

2.3.1 User-centric diary 19

2.3.2 Mobile sensing and activity recognition 19

2.3.3 Place of interest detection 20

2.3.4 Routine detection 22

3 Place of interest detection 25

3.1 Introduction 25

3.2 Clustering algorithm 26

3.2.1 ESOINN algorithm description 26

3.2.2 Improvements on the algorithm 30

3.3 Categorisation and representation of SEMs 33

3.4 Evaluation 33

3.4.1 Preliminary experiment 34

3.4.2 Clustering experiment 34

3.4.3 Performance evaluation 36

3.5 Conclusion 37

4 Routine recognition 39

4.1 Introduction 39

4.2 Generation of simple episodic memories 40

4.3 Generation of complex episodic memories 40

4.3.1 Finite state machine model 41

4.3.2 Frequent pattern model 44

4.4 Evaluation 52

4.4.1 Preprocessing 53

4.4.2 Generation of CEM 53

4.4.3 Comparison with T-patterns 56

4.4.4 Crowdsourced evaluation 57

4.4.5 Performance evaluation 59

4.5 Conclusion 61

II Glycemic State Recognition 63

5 Background 65

5.1 Introduction 65

5.2 General information 66

5.2.1 Diabetes 66

5.2.2 Insulins 68

5.2.3 Glucose monitoring 68

5.3 State of the art 70

5.3.1 Non-invasive glucose monitoring 70

5.3.2 Energy expenditure 72

6 Physiological model 75

6.1 Introduction 75

6.2 ECG alterations 75

6.3 Physiological Model 76

6.3.1 Adaptive filtering 76

6.3.2 mathematical morphology 78
6.3.3 QRS extraction 80
6.4 Evaluation 83
6.4.1 Dataset collection 83
6.4.2 Physiological model 84
6.4.3 Comparison with existing approaches 85
6.4.4 Results 86
6.4.5 Performance evaluation 86
6.5 Conclusion 87

7 Energy model **89**
7.1 Introduction 89
7.2 Energy model 89
7.2.1 Evaluation of the energy expenditure 90
7.2.2 Food intake modeling 94
7.2.3 Insulin intake modeling 96
7.2.4 Aggregation 97
7.3 Evaluation 97
7.3.1 Performance evaluation 98
7.4 Conclusion 99

8 Conclusions **101**
8.1 Outlooks 103

Bibliography **107**

Curriculum Vitae **119**

List of Figures

1.1 Various level of noise 3
1.2 Overview of the hierarchical framwork 5

2.1 The episodic memory model (adapted from [32]) 15
2.2 The architectural pipeline of MemorySense 16
2.3 Views of SEMs of MemorySense 18

3.1 ESOINN algorithm's flow, adapted from [50]. The modified steps are presented with grey background. 27
3.2 Cluster disappearance after 3 λ 31
3.3 Scaling factor mechanism (a.normal threshold b. scaled up threshold) 32
3.4 Influence of the scaling factor on the number of superfluous cluster 34
3.5 Edit distance between two words 36
3.6 Mobile performance of the modified ESOINN 37

4.1 Example of finite state machine. 42
4.2 Example of matching of a sequence of SEMs against an existing finite state machine. 44
4.3 Number of finite state machine created and number of unique boolean features combinations as a function of the number of days per user. 55
4.4 Crowdsourced experiment interface 58
4.5 Time required to process a new week of data for both SEM generation methods, compared to a non-incremental method based on original Apriori and GSP. . . . 60

5.1 Zephyr bioharness sensor belt 66
5.2 Insulin pump (Source: wikimedia) 69
5.3 State-of-the-art devices for blood glucose level monitoring 70

6.1 Fiducial points (PQRST) of a heart beat on an ECG representation. Adapted from Laitinen [71]. 76
6.2 Adaptive filter structure 77
6.3 ECG filtering using an adaptive filter fed with the accelerometer data 79
6.4 MM core operators 80
6.5 Structuring element and mathematical morphology operations on ECG 81

6.6 Comparison of the QRS detection between low and high entropy Signal 83
6.7 Processing time as a function of the length of the signal to process 87

7.1 Overview of the energy model 90
7.2 Breathing filtering using a low-pass filter 91
7.3 Evolution of the signals HR, BR, VMU as a function of the activity (in MET) . . . 93
7.4 MET of the 3 participants under various activities 94
7.5 Nutritional information as can be seen on fitbit 94
7.6 Conversion of the different components into glucose over time 95
7.7 Evolution of RMSE as a function of the number of entries per day 98
7.8 Variations in the accuracy of the model 99
7.9 Processing time to determine energy expenditure 100

List of Tables

3.1 ESOINN parameters . . . 26
3.2 ESOINN data structures . . . 26
3.3 ESOINN parameters for MemorySense . . . 33
3.4 Average precision and recall of the location clustering . . . 37

4.1 Dataset example . . . 51
4.2 Frequent itemsets . . . 51
4.3 Frequent sequences . . . 52
4.4 Boolean features ordered as: $\{home, work\}$, $\{home, restaurant\}$, $\{home \rightarrow restaurant\}$, $\{home \rightarrow work \rightarrow work\}$ . . . 52
4.5 Statistics of the dataset. Days per users and SEMs per day are defined by their average (μ) and standard deviation (σ) . . . 53
4.6 Statistics (averaged per user) for the frequent pattern matching approach . . . 55
4.7 T-pattern parameters . . . 56
4.8 Crowdsourced evaluation of T-patterns, FSM and FPM . . . 59
4.9 Confusion matrix for the classification tasks performed by the crowd (compound of T-patterns, FSM and FPM). All numbers are percentages, and the correct classifications are reported in bold. Rows are expected answers, columns are the input from the crowd. . . . 59

6.1 Features allowing to differentiate ECG under hypoglycaemia and euglycaemia . 76
6.2 Statistics on the data set collected . . . 84
6.3 Evaluation of the physiological model . . . 86

7.1 Root mean square error of the MET computation from averaged linear regressions 93
7.2 Pairwise RMSE between our model, the CGMS readings and manual glucometer measurements . . . 98

1 Introduction

Smart devices paved the way for ubiquitous computing and the paradigm of Internet of things. In this paradigm, heterogeneous devices are being endowed with the capacity to perform on-device computation, to communicate with other devices and to sense and interact with their environment. Technological advances of the past decade made it possible to miniaturise and to mass-produce such devices. In the application domain of smart homes, for instance, coffee machines, lights and thermostats can be activated remotely over WiFi, while security cameras can alert home owners upon unauthorised intrusion.

In the context of the Internet of things, sensors have become the counterpart of human senses for computing systems. They enable devices to prob the physical world and translate the collected information into the digital world. Examples of such sensors are embedded cameras monitoring road traffic, pollution sensors analysing the quality of the air in cities or fingerprint activated door locks. These sensors also go beyond what human can sense themselves consciously: For example proximity sensors, GPS sensors or electrocardiogram (ECG) sensors, which lead to applications such as map navigation or heart rate monitoring.

A subset of these sensors are embedded in wearable devices. These devices are worn by their user in order to sense their immediate surrounding environment and to log personal information. This latter usage led to the notion of quantified self: The possibility to track one's activity, performance or general state. This line of technology became popular with the physical activity tracking through wrist band devices which record various signals such as heart rate and accelerations. The data is then processed and can be presented to the user on her smartphone or directly on the device itself. This data is very sensitive in term of privacy and needs to be carefully processed.

Aligned with the general endeavour of improving the modelling and understanding of the different aspects of the reality by the machines, this thesis proposes to study how wearable devices can assist their wearer by automatically inferring their complex states. More specifically, we are interested in using commonly available devices to detect and recognise user-centric complex events. We consider these events as having the particularity of not being obvious and detectable directly from raw signals, but rather being built up in a hierarchical fashion and from multiple sensors. They are semantically richer compared to simple events, and provide the following advantages:

- They are easier to interpret for humans, providing the users with a bigger picture while giving them the possibility to look into each of these complex events to understand the underlying simpler events, the way one would zoom in on an image.
- They provide additional insights. Complex events leverage the information of underlying events to infer information that could not have been deduced by focusing on the simple events individually.

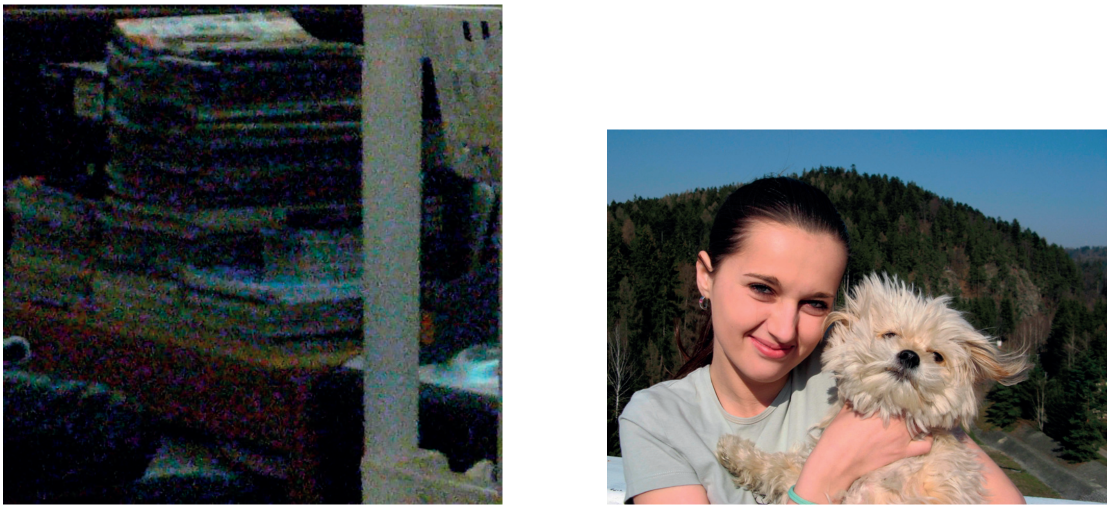
Complex events are of particular interest to the healthcare domain. Indeed, with the increase of mobile health devices such as smartphones with healthcare related applications, fitness trackers or connected medical devices, the amount of data that is available is enormous. However, processing these data on a large scale to produce actionable insights while ensuring the privacy of the patients is still not fully studied.

1.1 Challenges

The main objective tackled in this thesis is the recognition of complex events that are pertaining to the user. The majority of event detection strategies developed so far on mobile devices are designed for extracting simple meaningful information from various data signals. An example application is the detection of activity from accelerometers signal. In [117] and [121] the authors present rule-based approaches for events processing on mobile, but the complexity of the detected events remains limited. The recognition of complex events is still problematic in mobile environment. From this overarching task emerge challenges that need to be addressed. They are presented in the rest of this section.

1.1.1 Multi-level noise

As an adaptation to the analog and noisy physical world, human senses are accustomed to removing this noise and can also interpolate and extrapolate seamlessly missing data. The brain can for instance interpolate the blind spot from the eye by using peripheral information and the other eye. When listening to someone, the brain can filter out the moderate ambient noise. Similarly, sensors and smart devices have to cope with this noise at several levels. A first level of noise is present in the raw signals that are acquired by the sensors, which can be altered during the acquisition itself. Figure 1.1a conveys an example of noise on a picture captured by a digital camera. Such alteration can occur when increasing the gain of the sensor of the camera in order to increase its ISO sensitivity. This artefact makes it difficult for instance to distinguish the different documents in the middle stack. This type of noise can be handled using standard signal processing techniques such as applying a median filter.



(a) Low-level noise (source: Wikimedia)

(b) High-level noise (source: Wikimedia)

Figure 1.1 – Various level of noise

Another source of noise that is more specific to this thesis regards the complex events recognition. Complex events result from the processing of lower-level intermediary representations, e.g. simpler events. In these representations, some elements, although clearly defined, can be seen as noise and therefore need to be handled. In other words this noise can be seen as semantic noise: The noisy elements have a meaning but are interfering in the task at hand. To understand this higher level of noise we can borrow another example from computer vision. When performing human face detection in Figure 1.1b, an algorithm can detect the girl's face, but might also detect the dog as a contender for the face detection. The dog is not a noise in the sense that the CMOS sensor did not correctly capture it, but in the context of human face detection, the face of the dog presents similar features to the human face: two eyes, a nose, a mouth and can therefore be a candidate.

1.1.2 Privacy preservation

By design, wearable sensor data represents a facet of the user or of her environment. Such information is therefore very sensitive and need to be handled carefully. However, the relatively limited computing capabilities of smartphones compared to servers and cloud infrastructures, coupled with the fact that digital economy analysts agree that data is the new oil, led most services companies to rely on the cloud to store data and to perform heavy computation. Recently, privacy issues on the cloud have been brought to light ([120], [23]) as an ever escalating race between providers and attackers to protect (and respectively access), data on the cloud led to already multiple leaks of personal information such as passwords, emails, and pictures. These privacy concerns are especially important when dealing with health-related data, and several lines of research aim to allow information about user data to be used by external services while keeping the user anonymous. However, such approaches usually exhibit a trade-off between privacy versus utility, and increasing one will decrease the other. In order to address users concerns about what usage is made of their data, major actors of the market of mobile devices showed that it is possible to rely essentially on-device computation and limit interactions and data exchange with cloud services.. For instance, the health-related data collected by Apple Healthkit [10] are stored and encrypted locally, and a strict access control regulates the applications that want to read it. Another advantage of local processing is the ability to provide a service that degrades gracefully when an Internet connection cannot be established. Google translate [54] and Google navigation systems [55] are examples of such services, which can be queried offline.

1.1.3 Constrained resources

This thesis tackles another challenge: To perform as much computation as possible on device. Nowadays, smartphone can perform evermore complex tasks on-device thanks to multicore CPUs and several gigabytes of RAM memory (although not at the scale of servers). However, this computation is constrained by the battery capacity. Indeed, computation-intensive tasks drain the battery faster and should be carefully used. Especially when handling sensor data at very high sampling rate (e.g. an mobile ECG sensors can achieve a sampling rate of 250Hz), it is essential to be cautious about the complexity of the algorithms used as well as the frequency at which they are run.

1.2 Research directions

In order to study and address the challenges described in the previous section, as well as to provide a more concrete frame to this thesis, we present two different contexts related to quantified self and that use the two most prominent categories of wearable devices: Smartphones and sport tracker. Due to their respective set of sensors, Smartphone are more fit to study the mobility patterns and activities of the user, while sport trackers are better suited to study the user's physical state. We describe below each of the contexts.

1.2.1 Complex routine detection

In the last decade, smartphones became ubiquitous and capable devices to sense their environment and perform computation. Combined with the fact that users carry them at all time, smartphone became the wearable device-of-choice to study various patterns of the user. The most studied areas are activity recognition and localisation which make use respectively of accelerometer data and GPS signals. We present in the first part of this thesis a system called *MemorySense*. This system is based on human episodic memory and aims to recognise user's life events and represent them. This system builds upon standard recognition tasks to detect basic elements of the user's life and fuse these elements into a more abstracted view in a hierarchical fashion. We are creatures of habits and therefore we tend to follow circadian and

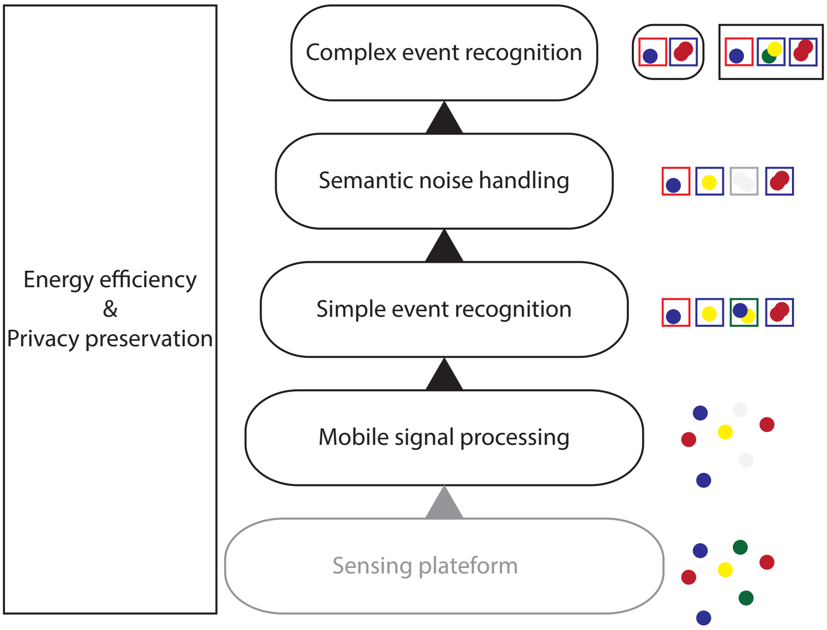


Figure 1.2 – Overview of the hierarchical framwork

weekly patterns. For this reason, we categorise these abstracted views into routines. While MemorySense could have a clear impact on memory impaired patients, it could assist a wider population by providing a digital diary of their past experiences.

1.2.2 Blood glucose level inference

The second part of this thesis focuses on the use of wearable sensors in healthcare, and more specifically in diabetes management. Sport devices contain sensors specifically focused on evaluating physical states of the user especially her level of physical activity. Multi-axial accelerometers are the core of sport device (e.g. Fitbit one [48, 38]). They provide an inexpensive way of sensing the motion of the user. Recently, additional sensors were included in these devices, such has GPS or heart rate monitor. Due to the increase in sensing capabilities, more advanced states can be detected using these sport devices. We present the system *D1namo* which aims to infer the glucose level of a user based on sport belt sensors. These sensors include accelerometers, electrocardiogram (ECG) and breathing amplitude. This approach aims at assisting an increasing amount of diabetic patients by proposing a non-invasive alternative to continuous glucose sensing.

1.3 Contributions

Each of the two directions proposes advances in its respective field. MemorySense proposes a system mimicking the human brain to represent a user's routines and life events, while D1namo presents a system for non-invasive blood glucose monitoring which can improve the quality of life of diabetes patients. However, these two systems are built following a common hierarchical framework for the detection of complex events on mobile devices which is presented in this thesis. This framework is presented in Figure 1.2, and it contains the following components:

Mobile signal processing Raw sensor signals are the base of every detection framework. They contain information about the user and her environment. However, these signals can exhibit noise resulting from the acquisition process. In MemorySense GPS signal can have a wide range of inaccuracy, in D1namo, the ECG signal can capture noise coming from friction of the electrodes. We will show in this thesis different techniques that can be used to alleviate this noise in an efficient way. These techniques will be designed so that they can be applied directly on the mobile device.

Simple event recognition Following the hierarchical approach of our framework, we then propose to detect simple events from the processed signals. In the case of MemorySense this translates into the detection of stay points where the user stayed for a sufficient amount of time to perform an activity. This is done using a modified version of an incremental clustering algorithm. In D1namo, we estimate the energy expenditure of patients based on the different belt sensors.

Semantic noise handling Different from raw sensor noise, semantic noise is defined by having semantic elements that are either contradictory with other or that do not fit the representation of the complex event that we want to model. In MemorySense for example, a user can have a clear "working day" routine in which she goes to work in the morning, goes to the work cafeteria at lunch time, goes back to work for the afternoon before coming back home in the evening. The fact that exceptionally she once stop by the supermarket before going back home should not prevent our system to classify her day as "working day". In this case, the supermarket event is considered as a semantic noise and should be handled. In the case of D1namo, the energy expenditure previously detected can be different depending on the sensor used to evaluate it. In public transport for instance, the acceleration may cause the accelerometer to yield a high energy expenditure, while ECG and breathing rate readings would remain normal and yield a basal energy expenditure. Discrepancy between these readings can be seen as semantic noise. For each context we present methods to handle this type of noise.

Complex event recognition The complex event detection comes as a natural result of the preceding steps. Once the semantic noise is detected, we present algorithms to process the other semantic elements and classify them accordingly. In the case of MemorySense we present two algorithms based on frequent pattern mining and finite state machines to recognise the different routines of the user and classify her days into one of these routines. In D1namo, we present two methods to detect the event of low sugar level in the user's blood. The first method is based on physiological symptoms of low sugar level in the electrocardiogram signal, the second method compares the offer of sugar in the blood compared to its demand in order to determine at any time the current sugar level.

Privacy preservation and energy efficient computation To address the privacy concerns that could arise from sensitive data being sent remotely, we propose to perform each stage of the framework on device. However, mobile devices have limited resources in term of battery and computational power. As a result, we propose algorithms that are suitable to be run on mobile devices.

1.4 Thesis structure

This thesis is organised in two parts, each one covering a different research direction presented above. In each of these parts, we will see how the hierarchical framework can be applied in order to detect complex events.

The first part covers the work done on the detection of complex mobility patterns. In Chapter 2 we introduce the key material pertaining to memorySense, in Chapter 3 we present our efforts on building the foundation for complex events detection, by detecting simple life events. In Chapter 4 we describe how these simple events can be leveraged to create more complex representations of the user's life.

The second part of this thesis covers the work done on the detection of blood glucose level based on non-invasive sensors. In Chapter 5, we provide the background knowledge that will be reused through the part, In Chapter 6, we present a solution to the recognition of hypoglycaemic events solely based on non invasive mobile sensors, and in Chapter 7 we leverage additional data sources and propose an alternative approach.

Finally, Chapter 8 concludes this thesis by providing a summary of its achievements, and proposes new research directions.

1.5 Publications & contributions of the author

This thesis is based on the work presented in the following publications:

part I

- Aberer, Karl, et al. "Memorysense: Reconstructing and ranking user memories on mobile devices." *Pervasive Computing and Communications Workshops (PERCOM Workshops)*,2014 [1]. In this publication, the author was jointly responsible for the design of the MemorySense vision with Dr Catasta, Dr Vasirani and prof. Aberer, and was responsible for the enhancement of the ESOINN algorithm and its evaluation. Dr Catasta was responsible for setting up the Nokia data challenge dataset. This work was supervised by Prof. Aberer.

- Ranvier, Jean-Eudes, et al. "RoutineSense: A mobile sensing framework for the reconstruction of user routines." *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2015 [95]. In this publication, the author led the design and implementation of the two approaches to routine recognition with the help of Dr Vasirani and Dr Catasta. The author was responsible for the evaluation of the system using the dataset setup by Dr Catasta. He also designed and implemented the task of the crowdsourced experiment that was run on the crowdsourcing platform by Dr Catasta. This work was supervised by Prof. Aberer.

- Ranvier, Jean-Eudes, et al. "MEmoIt: From Lifelogging Application to research platform." *Mobile Application Development, Usability, and Security* (2016): 1 [94]. In this publication, the author was responsible for aggregating and presenting the work that was performed on MemorySense from an engineering perspective by the mobile application which implementation was led by the master students Ivan Gravilovic, George Christodoulou, Horia Radu and Tiziano Signo', and from a research perspective regarding place of interest detection and routine recognition.

part II

- Ranvier, Jean-Eudes, et al. "Detection of hypoglycaemic events through wearable sensors." *Proceedings of the International Workshop on Semantic Web Technologies for Mobile and Pervasive Environments*, 2016 [96]. In this publication, the author was responsible for the design of the d1namo architecture, and building the processing pipeline necessary for building the physiological model to detect hypoglycaemia. Regarding the collection of the dataset used in this paper, Fabien Dubosson was responsible for the application to the ethical committee for the data collection of diabetes type 1 patients, and the author was responsible for the relation with the medical staff and the collection of the data. Dr Calbimonte was responsible for the semantic integration part of this publication. This work was supervised by Prof. Aberer.

- Ranvier, Jean-Eudes et al. "Non-invasive Detection of Hypoglycemia for Type 1 Diabetic Patients" submitted to *Computers in biology and medicine*, 2016. In this publication under submission, the author was responsible for the evaluation of the physiological model, the design and the implementation of the energy model to estimate the blood glucose level, and its evaluation. This work was supervised by Dr Calbimonte and Prof. Aberer.

Sensing user memories Part I

2 Background

2.1 Introduction

The amount of digital information produced by users is massive. Thanks to an ever-increasing range of hardware devices whose capabilities soar while their price plummets (e.g., smartphones, smartwatches), the digital trails of the user behaviour, lifestyle, and social relationships are scattered across a collection of unstructured repositories. As such, efficient long-term archival and querying strategies are very difficult to devise. How to scavenge this data, organise it, extract meaningful information, and present it to the user is of paramount importance to tap into the chaotic piles of digital information that the users (semi-consciously) dump online.

Departing from classical infrastructures that provide static views over heterogeneous sources, we propose to mimic the way our brain stores and accesses information [32] in order to narrow the cognitive burden of accessing digital memories. We focus on collecting and analysing mobile data from smartphones, using multimodal data fusion, semi-supervised online classification, and personalised statistical models to automatically extract user digital memories. In MemorySense, we aim at leveraging the sensors embedded in modern smartphones (i.e., GPS, accelerometer) and combining with external data sources (i.e., Google Places API, Foursquare API) to answer the questions that are at the root of the creation of memories (*what* occurred during an episode, *where* the episode took place, and *when* the episode happened [32]). The development of MemorySense on a mobile platform poses unique challenges: coping with the limited computational power, battery; filtering noise from the input data; reconstructing memories from partial chunks of information; ranking the memories according to the user preferences, and representing them in an appealing form. Although part of the resource limitations could be overcome by migrating the computations to a cloud infrastructure, Out of concerns for the user's privacy, MemorySense has been designed to limit exchange of sensitive user data with third-parties.

2.2 Global vision

2.2.1 Episodic Memory Model

Defined by Turving in 1942, the episodic memory represents the human system responsible for storing and recollecting past experiences (e.g., had dinner with Alice last Saturday). This system is to be put in parallel with the semantic memory, responsible for the storage of structured record of facts (e.g. Alice has two siblings, Bob and Cathy).

In [32], Conway theorises the way in which episodic memories are built, stored and retrieved while preserving their temporal dimension. He defines three elements that are combined hierarchically to build episodic memories: Episodic elements (EEs), simple episodic memories (SEMs) and complex episodic memories (CEMs).

Episodic elements are the most atomic elements of the episodic memory model. They represent events or summary of events and can be seen as details of an experience. They are often associated with visual cues of such experience. EE are usually attached to a conceptual frame that provides a context. Such frame can regroup one or more EE. In some cases, the frame of an EE can be missing leading to the concept of free radicals. Free radicals are EEs without a frame and therefore without context. An example of such free radicals is the memories associated with the jokes. Jokes do not necessarily need a context and can be memories "on their own".

Still according to Conway's model, episodic elements associated with their frame form a group names simple episodic memories (SEM) (c.f. Figure 2.1). This group represent a short episode, such as *commuting to work*, and contains details about this experience. It can be accessed though its contextual frame.

The last component is the complex episodic memory. Multiple SEM can be organised into a CEM by being aggregated inside a common conceptual frame. This frame is more abstract than its EE counterpart, and provides a context for the CEM. CEMs have typically a longer time span than SEMs and can cover life episodes such as *a day at work*. During this complex event, several simpler SEMs will have happened such as *going to work*, or *having a coffee break*.

We propose to translate Conway's model into the digital world. We saw in the introduction of this thesis that there is a strong analogy between human senses and digital sensors. In MemorySense we push this analogy forward by developing a framework for logging episodic memories based on the way the human brain stores them. This approach has the advantage of limiting the cognitive burden of new users as they are already implicitly familiar with the model, while offering an efficient way to look-up memory through their EEs. We begin our framework by assimilating the concept of EE to a signals or an information related to event. Example of such EE can be a GPS location or a phone call. We then define SEM, which represents the gathering of one or more EEs and their associated frame as a 3-dimensional entity containing the answers to the questions *When?*, *Where?* and *What?* (e.g. At 10am I went jogging by the lake

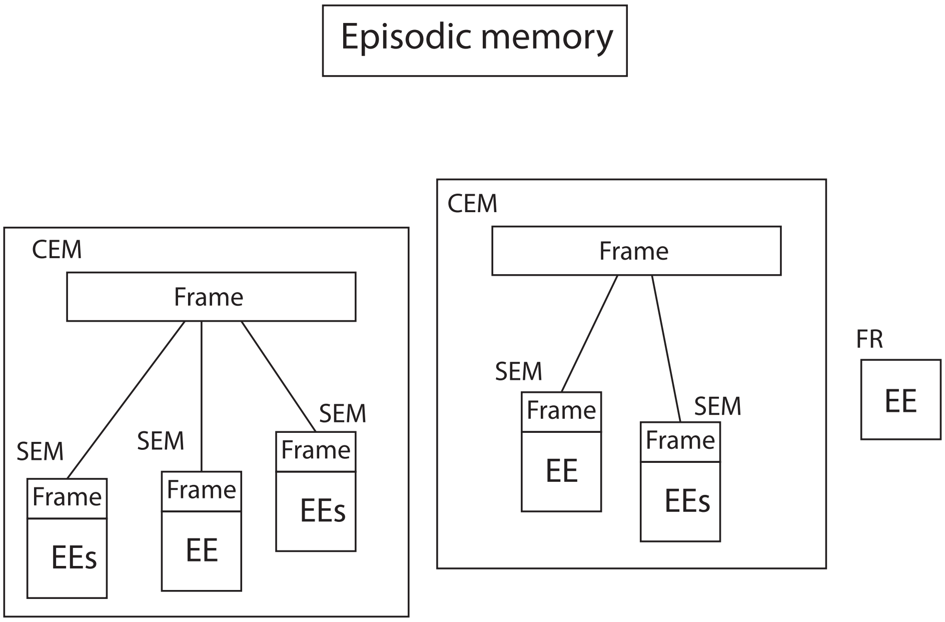
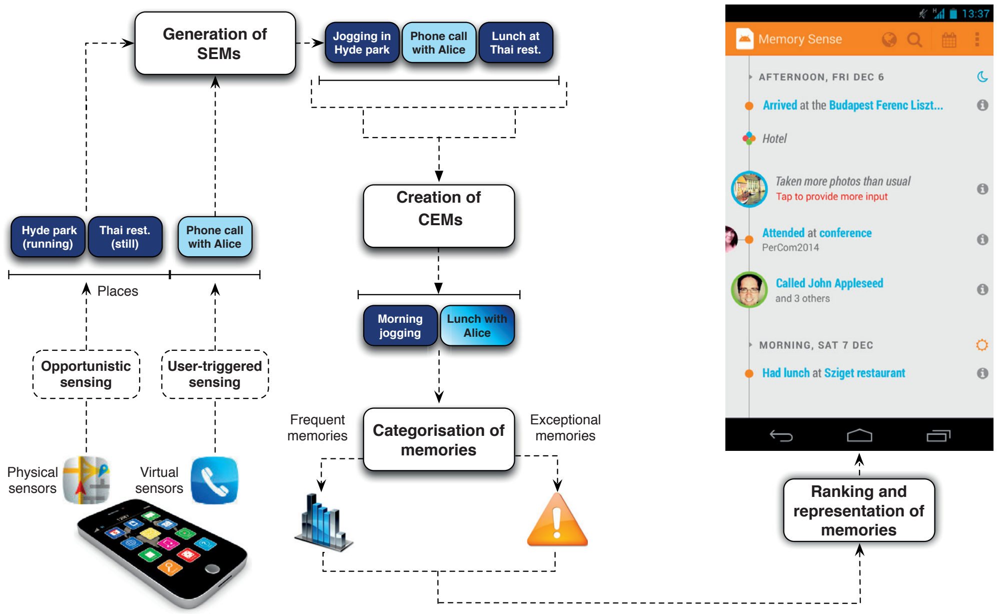


Figure 2.1 – The episodic memory model (adapted from [32])

side). The *when* relates to the temporal nature of the memories and correspond to the instant at which an event occurs. It is naturally represented in our system as the timestamp at which this event happened. The *where* corresponds to the location at which the event happened. While GPS coordinates would be a natural candidate to represent this dimension, they lack interpretability. We propose to augment these coordinates with a semantic representation of the venue. The *what* represents the activity being carried out by the user. The representation of the activity also follows a semantic approach setting the base for comparison and querying of activities. Each of these 3 dimensions is an EE. However, the semantic interpretation of the *where* and *what* provide the contextual information composing the frame of the SEM. Finally, we emulate the notion of CEM in our system via 2 mechanisms. By 1/ allowing the user to manually group SEMs to create CEM and 2/ automatically discovering recurring patterns of SEMs in the past activities (e.g. going to a restaurant and then going to the opera can be aggregated as going on a date). We assimilate CEMs to the notion of routines.

This approach has 2 main benefits:

- The cognitive burden while interacting with the system is reduced, hence facilitating higher memory recall (i.e., MemorySense acts like a "memory prosthesis")
- SEMs and CEMs encourage composability of memories (i.e., we interpret SEMs as atomic events, and SEMs as high-level overviews of the events happened during a longer timespan).



AFTERNOON, FRI DEC 6

Arrived at the Budapest Ferenc Liszt...

Hotel

Taken more photos than usual
Tap to provide more input

Attended at conference
PerCom2014

Called John Appleseed
and 3 others

MORNING, SAT 7 DEC

Had lunch at Sziget restaurant

Figure 2.2 – The architectural pipeline of MemorySense

2.2.2 System architecture

MemorySense generates a high-level representation of a user's life based on his smartphone sensor readings. Inspired by the episodic memory model proposed by Conway, we structure our framework based on his representation of the human memory responsible for storing and recollecting past experiences. Figure 2.2 depicts the resulting approach taken by MemorySense. First, the SEMs are generated by an algorithm for place of interest detection, and by the virtual sensors in the phone (e.g., phone calls, SMS, etc.) Then, the routines are reconstructed directly from the SEMs. Finally, our prototype application, MemorySense, displays the routines at different granularity levels on a timeline.

Generation of SEMs

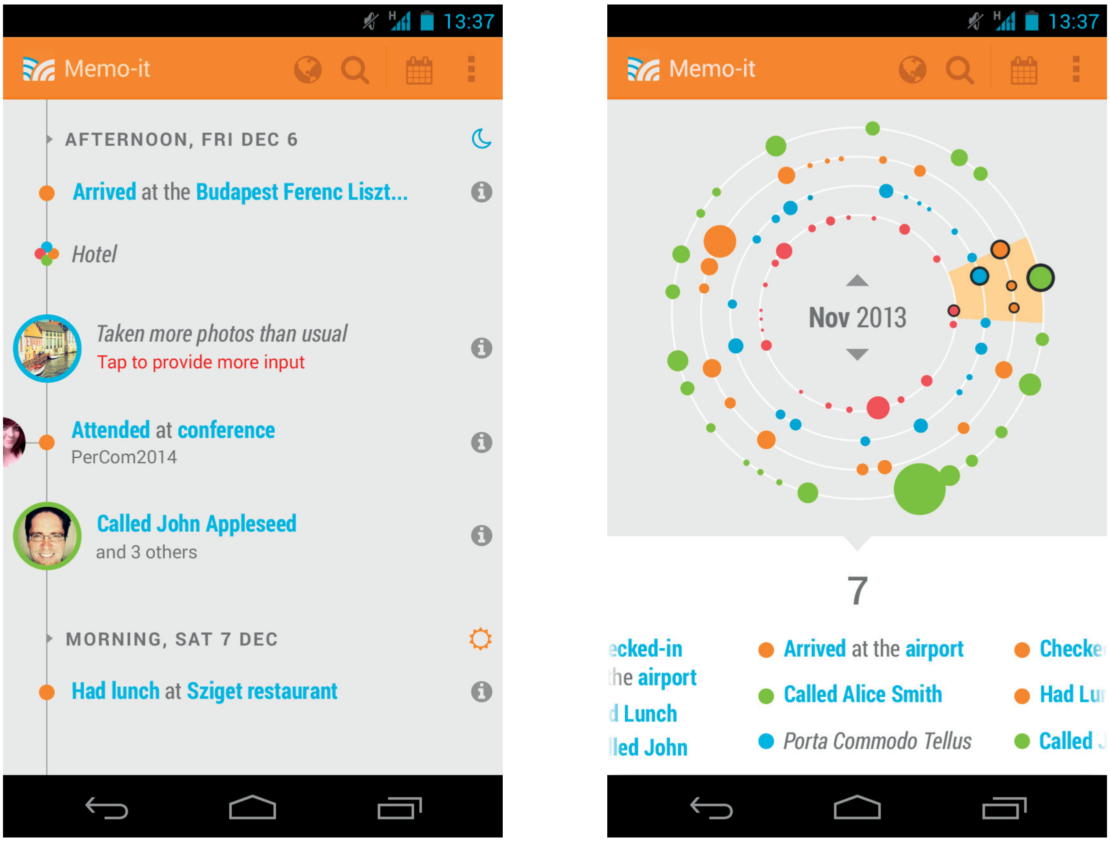
MemorySense defines two types of SEMs.

The first one is related to the opportunistic sensing of the user's whereabouts. Opportunistic episodic elements contain the following information: the time of the activity, the place of interest at which the event occurs and the description of the activity the user is performing. The time of the activity is represented by the timestamps at which the activity started and ended. Places of interest are defined as the locations at which the user stayed for a minimum amount of time to perform an activity. They are generated based on a GPS data to determine the location of the user. Due to the imprecision in GPS readings and the continuous nature of the spatial domain, stay points need to be clustered in order to detect recurrent visits of the same place. The clustering allows as well to avoid having to characterise a new place of interest if it belongs to an already characterised cluster. This problem is handled by using the ESOINN algorithm [50] to cluster stay points. ESOINN possesses several suitable features to this purpose such as being incremental, unsupervised and does not require to know a priori the number of clusters. The output of the algorithm represents the unique places visited by the user. However, some modifications were required to better fit the needs of MemorySense. These modifications include the removal of the forgetting mechanism of ESOINN in order to retain all the data, modification of the density function in order to simplify the computation, therefore making it suitable for mobile computing, and generation of consistent cluster labels to reuse from one increment to another. Once the stay points are acquired and clustered, we enrich them semantically by attributing a venue to each cluster. The attribution is done by querying location-based services such as Foursquare or Google places to get a list of the k-closest venues. The description of the activity is composed of a verb and a complement. In order to semantically structure the description, verbs and complements are selected from the Wordnet database ([83, 47]). This approach allows for consistency in the descriptions and facilitates the definition of a distance between two description. As for the stay points characterisation, MemorySense makes use of the history of the user's activities to generate a list of potential descriptions based on the frequency of activities at the specific places of interest.

The second type of SEM generated by MemorySense is based on the virtual sensors of the phone. They represent the interaction that the user had with the phone and are therefore called user-triggered. They encompass calls and sms received and given by the user, calendar events, pictures taken and social interactions. Some of these SEMs are not necessarily geo-localizable (i.e. sms, calls) and therefore it would be difficult to attribute them a meaningful location. However, they all contain a timestamp and an activity, which allow us to fuse them in the timeline, therefore, diversifying the information collected about the user.

Generation of CEMs

According to Conway, Complex episodic memories represent an abstraction of SEMs. In MemorySense we translate this concept by introducing the notion of routines. Routines are defined as sequences of activities that the user performs. These routines can be frequent (e.g. standard working day) or exceptional (e.g. the wedding of a friend). Given the infinite amount of activities that can be performed by a user, we propose two approaches to the modelling of routines without setting restrictions and the type of activities that can be included in these routines. The first model is based on a finite state machine approach, and generates a graphical model of routines which can simulate all the specific CEMs that are associated to the routine. The second approach, instead, wants to capture what are the SEMs that make a routine unique and what are their relationship with each other. This approach is based on frequent pattern mining.



AFTERNOON, FRI DEC 6

Arrived at the Budapest Ferenc Liszt...

Hotel

Taken more photos than usual
Tap to provide more input

Attended at conference
PerCom2014

Called John Appleseed
and 3 others

MORNING, SAT 7 DEC

Had lunch at Sziget restaurant

ecked-in
he airport
d Lunch
lled John

Arrived at the airport
Called Alice Smith
Porta Commodo Tellus

Checke
Had Lu
Called J

(a) Timeline view (b) Semantic view at a month granularity

Figure 2.3 – Views of SEMs of MemorySense

Representation of memories

In MemorySense, memories are shown to the user in a graphical user interface. According to Conway, the episodic memories can be represented by the brain in a temporal dimension and in order of appearance. We model this approach by developing a first view consisting of a scrolling storyline of the episodic memories and free radicals of the user. with the possibility for the user to dive into a more specific period. A coarse-grained version of the storyline (Figure 2.3a) should display only the most important (in the sense defined in the previous subsection) memories, while the zooming function should provide the user with memories with a lower ranking as well. A second view (Figure 2.3b) focuses on the meaning of the episodic memories, by grouping semantically-related memories together. This relation can be computed using the activity attached to the episodic memories.

2.3 State of the art

2.3.1 User-centric diary

Collecting documents and information about a user while offering aggregation and visualisation capabilities over such data is a well-explored area. In 1945, at the genesis of the computer era, Vannevar envisioned Memex [20], a hypothetical system in which users would store all the documents related to their lives. In the early 2000, the MyLifeBits project [52] aimed at fulfilling the Memex vision by developing a system that stores all the digital media of a user, offering visualisations and annotations capabilities. This system would allow not only to store different sources of information, but also to search the database through full text search, annotations and hyperlinks. Czerwinski et al ([34]) explore the memorability of events occurring while operating a computing system, showing, among other things, that recurring events tend to be naturally forgotten. Hodges et al. present Sensecam ([64]) to record visual cues along with sensors data to improve the recall of amnesic patients. In [24], the authors propose to build a desktop search system based on the observation that humans tend to work by association of ideas and attend to leverage this idea to improve retrieval of documents.

More recently, with the popularisation of mobile platforms, and thanks to the increasing number of embedded sensors such as GPS, accelerometer and microphone, user-centric sensing applications built on smartphones and wearable devices are enabling new ways to sense and understand the way we live (e.g., quantified self) and the world that surrounds us, motivating several research directions. In [16]), Bicocchi et al. propose a whereabout diary based on GPS coordinates to build user's trajectories and her places of interest. The authors rely on heuristics to extract points of interest and then augment them with semantic information using third party services. Closely related, in [46], the authors develop a diary containing the stay points of the users collected from smartphones and augmented with external information from Google map and Yelp in order to allow for searchable locations. In [58] and [57], Guo et al. propose to harness mobile sensing in order to improve the recall of human memory. To this purpose, they extract contact information and memory cues from physical and virtual sensors and structure this contact information in order to facilitate its recall through the appropriate cues. However, all these operations are usually performed on single or pairs of activities. More specifically, by using mobile tagging, such as RFID or barcodes, human memories are associated with physical objects, effectively externalising part of the user memories, while at the same time building an object-based social network for memory sharing.

2.3.2 Mobile sensing and activity recognition

There exist a large amount of work that focus on structuring the digital information and recognising/quantifying specific aspects such as the physical activities or the visited places based on various embedded sensors. In 2008, Choudhury et al.[[27]] presented a customised mobile sensing platform to detect user activity. The approach leverages standard sensors that can be found in smartphone as well as innovative ones such as barometer, humidity and infrared light sensor to infer activities of the user. The field study they performed allowed distinguishing between various physical activities. The authors also evoke concerns about privacy of the data and emit the idea of performing computation on device. Yi He [61], proposes to add fuse accelerometers, gyroscope and compass to detect a set of physical activity based on a classifier hierarchy. With similar purpose, but solely using accelerometers data, Yan et al. in [123], Ravi in [97] and Kwapisz et al. in [70] also infer various activities of the user. In [14], the authors present SurroundSense, a multimodal approach to localisation by using the ambient noise, brightness, color, wifi fingerprint and even accelerometers to logical location of the user. Instead of yielding absolute GPS coordinates of the user's location, the system detects if the user is in an environment previously visited. Lu et al. in [78] present a system based on the microphone to detect ambient sounds, music and speech. From this detections the authors build a context classifier to label the context of the user at a specific time (e.g. the user is walking outdoor, walking indoor, driving, etc). In [130], Zheng et al. describe a supervised learning approach to augment a timeline of raw GPS coordinates with the different inferred transportation modes. This system is part of a larger project [131] which proposes storage and search mechanism for these augmented GPS trajectories. The transportation mode detection problem is also studied in [101]. Finally, Peebles presents in [91] an approach to activity detection using labeled activities from multiple users.The labels, input as free text by the users are robustly disambiguated to enable to allow the system to group similar events despite them having different labels.

Another line of work concerns the ease of access of sensors data and the energy efficiency of data acquisition. In [79], the authors present a sensing framework to allow continuous sensing of multiple mobile sensors (GPS, accelerometers, microphone) and propose a first layer of processing of the sensor data to recognise various events and activity such as human voice detection, activity recognition. Similarly, in [119], the authors explain that the energy-hungry nature of sensors is a major issue of mobile sensing platforms. They devise new strategies to limit the energy footprint of a multi-sensors framework for the recognition of user states. Another example is Funf, an open sensing framework [5], which aims at helping developers to build their own sensing applications by providing an standard API to access mobile sensors. Ubiqlog [99] also present a general purpose system to facilitate life-logging on mobile devices. Finally, Agarwal et al. in [2] propose a middleware enabling mobile developer to create sensing application, and perform this sensing on large scale.

2.3.3 Place of interest detection

Smartphones gave to GPS, and localisation mechanisms in general, a new purpose. However, when used, GPS sensors drain the battery significantly. Despite technological improvements at the sensor level, such as the Assisted “A-GPS”, continuously acquiring the precise localisation information, is generally costly. Several approaches have been taken to solve this problem, such as adapting the rate at which the GPS sensor is activated ([90]). The Android ecosystem abstracts this problem by proposing a geofencing API that handles the calls to the GPS sensor.

In addition to being able to know her exact location, the user also gains access to additional location-based services. One of them is the detection of places of interests (PoI) as described in [85]). Places of interests are defined as geographical regions having a semantic meaning (e.g. a cinema, a restaurant, a landmark), where the user stayed for a certain amount of time. This definition is instrumental in the comprehension of users' behaviours at a semantic level. An application of this abstraction is the recommendation of PoI ([15]) or the check-in service of Foursquare ([49]). In [122], the authors propose a framework for annotating trajectories of various objects, structuring these annotations with semantics.

The detection of PoI necessitate to be able to cluster GPS locations together in order to 1. detect the places where the user spend a certain amount of time and therefore is not transiting, and 2. be able to recognise that a certain GPS location is the same than a previous visited location, although the GPS coordinates are not exactly equal. Montoliu et al. study two different algorithms for that purpose:

- DBSCAN, introduced by Ester et al. ([43]) is a non-incremental density based clustering algorithm. Starting from any datapoint available, the algorithm tries to grow clusters from dense areas and removing the noise in between these clusters. This approach has the advantage of not requiring the number of clusters before-hand. this allows for an arbitrary number of stay points.
- A grid-based clustering algorithm presented in [129]. This algorithm starts by discovering stay points where a user will have spent more than a time $T_{threshold}$ and during which any two GPS coordinates are no further than $D_{threshold}$, the algorithm then generate a grid on top of the map and cluster grid cells together in such a way that the maximum size of a cluster is not bigger than a predefined parameter d.

An alternative density-based clustering algorithm that can also be found in the literature for location clustering is OPTICS. It is used in [9] to detect meaningful locations. The clustering algorithm is sensibly similar to DBSCAN, with the difference that it handles variation in the density of the different points.

Another line of algorithm is based on using clustering algorithm for convex shapes such as K-means, or the approach taken by Kang et al. in [68]. K-mean, used in [12] in the context of location clustering, initially defines k centroids within the data space and then iteratively adjust these centroids to achieve optimal clustering. This algorithm has several disadvantages such as requiring a predefined number of clusters (i.e. places of interest that can be detected), not being incremental as well as potentially converging toward a sub-optimal cluster configuration. The algorithm proposed by Kang et al. incrementally builds clusters based on the distance between consecutive GPS data points. The fix distance parameter makes it unpractical for a dense area with close points of interest.

2.3.4 Routine detection

Bounded-States Routine Detection

The aggregation of simple activities into high-level routines has also been explored in previous works. In [26], the authors propose a method to discover complex activities based on smart home sensors. They use a latent Dirichlet allocation(LDA) and frequent sequence mining algorithm to build complex activities. In this context, the number of sensors and the number of activities that can be tracked are limited. This limitation allows the authors to fix beforehand the number of topics used in LDA. Similarly, Nguyen ([87]) discovers complex patterns using hierarchical Dirichlet processes. In [17], the authors also aim at detecting a finite set of activities based on accelerometer data. The K-mean algorithm is used first to cluster sensors data into low level events which are then processed by a boosted set of classifiers to determine which high level activity is performed. However, K-mean necessitates the a priori knowledge of the number of events that can be detected, making this method not suitable for problems in which the number of states or activities is unbounded. The authors of [45, 44], and [40] use a similar representation of states (home, work, out, no information) to perform a 4-class classification. Farrahi and Gattica-Perez, in [45], segment the user's location trace into a time series of labels before modelling it using LDA augmented with the author-topic model, to define routines. Conversely, Eagle et al. [40] capture the behaviour of users using the notion of *eigenbehaviors*, the principal components of a behavioral space. Once again, these approaches are restricted to the detection of a finite set of states. While combinations of these states can lead to multiple routines, it does not allow for a fine grain distinction between them.

Unbounded-States Routine Detection

Ye et al. [125] leverage GPS traces of the user to detect temporal and conditional life patterns using non-incremental association rule mining. The authors focus on the prediction of future life trends of the user and evaluate the trade-off between representativeness and interestingness of temporal patterns. However, they do not study the use of the discovered trends to classify days into different routines. Coming from the neuroscience domain, Magnusson et al. ([80]) present T-pattern as a mean to study human and animal behaviors. The authors build a hierarchy of temporal events that are seen repeated in a stream of events. This approach does not offer the necessary flexibility to model routines with variations such as swapping of events or missing events. The notion of T-pattern presented in [53] is different from the T-pattern presented by Magnusson. Giannotti describes frequent behaviours both in term of space and time. T-patterns are mined from GPS traces by first establishing a list of dense regions traversed by the users. These regions are then analysed from a temporal perspective in order to extract frequent temporal patterns. This method accepts an unbounded number of states and can generate an infinite number of patterns. However, this method focuses on the GPS representation of the data and therefore does not benefit from additional semantic information. Furthermore, it focuses on sequential patterns, disregarding potential temporal swapping of events.

The same way the human brain structures memories in a hierarchical fashion, we propose a framework which aggregates low level sensor readings into more abstract multimodal states which can be, in turn, aggregated into high level representation of the user routine. It improves upon existing work by allowing for the discovery of an unbounded set of states defined by their semantic representation and which are then aggregated into an unbounded set of routines. The algorithms used by the framework are essentially incremental. This allows for periodic processing of the data increments as well as energy efficiency, which are both required for mobile applications.

3 Place of interest detection

3.1 Introduction

In the overview of MemorySense, we distinguished the SEMs coming from virtual sensors such as phone calls and SEMs coming from the opportunistic sensing. This sensing introduces a first layer of noise implied by the inaccuracy of the GPS sensors. This inaccuracy makes it challenging to properly extract the trajectory of the users and to understand where the user performed activities.

In this chapter, we present the generation of the SEMs based on the opportunistic sensing of the user's activities as proposed in MemorySense. We are interested in the places of interest (PoI) of the user, which are defined as the locations where the user stayed for a minimum amount of time to perform an activity. We leverage the Android activity recognition API, which provides an easy way to detect if the user is traveling (i.e., in a vehicle, on a bicycle, etc.) in order to de-activate the energy-consuming GPS sensor, therefore saving energy while applying a first step towards filtering the data and extract PoIs.

However, the GPS coordinates provided by the location sensing are given as real values and therefore need to be clustered for three reasons: (i) to distinguish whether the user stays within an area considered as a unique location; (ii) to alleviate the inaccuracy of the GPS sensor; (iii) to limit the computational cost of determining the location type by avoiding the categorisation of GPS coordinates belonging to an already classified location.

We use the clustering algorithm ESOINN, described in [50], which provides the following advantages:

- **Online** It does not require the entire dataset before starting classification
- **Incremental** It does not have a specific training phase and can train with new data as they arrive
- **Unsupervised** It allows clustering signals without the use of ground truth.

- **Unknown number of clusters** Some clustering algorithms require the number of clusters as a parameter. ESOINN allows dynamic detection of this number.
- **Density based** Density-based algorithms are more flexible regarding the shape of clusters.

3.2 Clustering algorithm

In this section, we present the original ESOINN algorithm and we describe the improvement that we made.

3.2.1 ESOINN algorithm description

The general flow of the algorithm is presented in figure 3.1. The gist of the algorithm is to transform input signals into networks of (fewer) nodes based on the density distribution of the input data. ESOINN requires the parameters defined in Table 3.1. The parameters are used to regulate the size of the network. The algorithm also uses several data structures, especially a complex representation of the node of the network. These data structures are summarised in Table 3.2.

Parameters	Description	Default Value
age_{max}	Age after which an edge will be deleted	100
λ	Period triggering a sanitation processing	100
c_1, c_2	Node deletion control parameters	0.001, 1.0

Table 3.1 – ESOINN parameters

Structure	Attribute	Description
Input signal	$features$	represents the signal (e.g. GPS coordinates)
Node	$features$ $subclass$ $points$ $periodPoints$ $distinctLambda$ h	represents the signal (e.g. GPS coordinates) subclass to which belong the node number of input signal represented by the node input signals attributed in the current λ distinct λs when the node has been updated Local density at the node
Edge	age	current age of the edge

Table 3.2 – ESOINN data structures

The algorithm necessitate a notion of distance between points (input signal or node) in the features space. Given the nature of our data (GPS coordinates), we use the haversine distance, which computes the shortest distance between 2 points on a sphere, as our standard distance measure. Given two points P_1 et P_2 defined by their GPS coordinates in radian $P_1 = (lat_1, lon_2)$ and $P_2 = (lat_2, lon_2)$, the haversine is given by hav in Equation 3.1. With R the radius of the

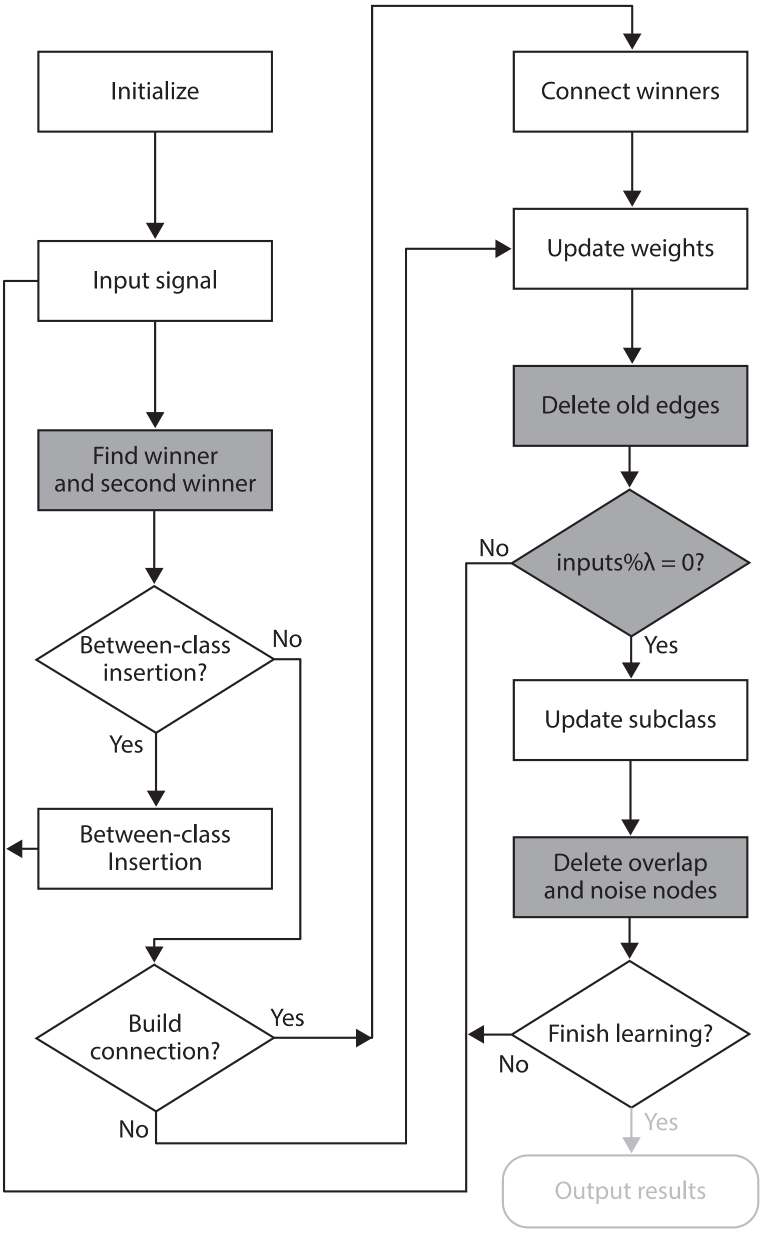


Figure 3.1 – ESOINN algorithm's flow, adapted from [50]. The modified steps are presented with grey background.

Earth (6371km).

$$hav(P_1, P_2) = 2 * R * atan2(\sqrt{a}, \sqrt{1-a})$$

$$\text{with } a = sin\left(\frac{lat_2 - lat_1}{2}\right)^2 + cos(lat_1) * cos(lat_2) * sin\left(\frac{lon_2 - lon_1}{2}\right)^2 \tag{3.1}$$

Algorithm flow

Signal insertion To process a new input signal x, ESOINN starts by determining its two nearest nodes (w_1 and w_2) in the network based on the haversine distance. It then evaluates if x belong to one of the two nodes or if it should generate a new node based on the similarity thresholds of w_1, w_2. The similarity threshold of node n is presented in Equation 3.2, with $N(n)$ representing the nodes directly connected to n, and N all the nodes of the network.

$$T_n = \begin{cases} \max\limits_{m \in N(n)} (hav(n,m)) & \text{if } n \text{ has any connection in the network} \\ \min\limits_{m \in N} (hav(n,m)) & \text{otherwise} \end{cases} \tag{3.2}$$

if x is within the threshold T_{w_1} or T_{w_2}, it will contribute to increase the density of their respective node and updates their coordinates, otherwise it will generate a new node in the network and finish the algorithm.

Network management Once x is inserted in the network, a series of updates is triggered. ESOINN beginnings by incrementing by 1 the age attribute of the edges connected to w_1. It will then decide if an edge should be present between w_1 and w_2. This decision depends on the $subclass$es of the two nodes and is particularly important when both nodes lie at the overlap between two clusters. An edge between w_1 and w_2 is created in one of the three following cases: 1/ One of the two nodes is recent and has not been attributed a $subclass$ yet (this attribution is triggered every λ signals). 2/ Both nodes belong to the same subclass. 3/ the two nodes are from different subclass but the minimum of their density h is above a threshold depending on the average density of their neighbourhood. If an edge is created between two nodes that are already connected, the age of this edge is set to 0.

Instead of computing the density at a node as the number of signals which found this node as being the closest, the density of a node is instead defined as a function of the mean distance ($\overline{d}_i$) from this node to its neighbours (m). From Equation 3.3 we see that the denser the underlying signals are, the smaller the $\overline{d}_i$. The authors define then a quantity p_i which is a function of $\overline{d}_i$ as defined in Equation 3.4. Finally, the density of a node i (h_i) is defined in Equation 3.5, where P corresponds to the number of period λ when p_i was not *null* and n represents the total number of inputs. This atypical definition of the density has the advantage to keep nodes in area not often activated to prevent their density value from decreasing.

$$\overline{d}_i = \frac{1}{m} \sum_{j=1}^{m} dist(node_i, node_j) \tag{3.3}$$

$$p_i = \begin{cases} \frac{1}{(1+\overline{d}_i)^2} & \text{if } i \in 2 \text{ nearest neighbours} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$\overline{h}_i = \frac{1}{P} \sum_{i=1}^{n} p_i \tag{3.5}$$

Next, ESOINN updates the $points$, $periodPoints$ and $distinctLambda$ variable of w_1 to reflect that it has assimilated x. The algorithm also updates the feature vector of w_1 and all its neighbours w_n to take into account the feature vector of x. The update is described in Equation 3.6. Finally ESOINN scans the network to delete edges e for which $age_e >= age_max$.

$$\begin{aligned} \Delta features_{w_1} &= \frac{W_x - W_{w_1}}{points} \\ \Delta features_{w_n} &= \frac{W_x - W_{w_n}}{100 * points} \end{aligned} \tag{3.6}$$

Sanitisation On a regular basis (every λ signals) a cleaning task is performed to remove unnecessary edges and nodes and to distinguish overlapping clusters. This distinction is determined by the underlying density of the nodes. The algorithm checks for composite clusters (i.e. with a density distribution containing more than one local maximum). To handle this problem, the algorithm first detects the apexes (node with a local maximum density) and attributes a different subclass label to each of them. Label is spread to all nodes connected to the apex. In the case some nodes are in an overlapping area between two apexes, the algorithm checks if the minimum of the density of the overlapping nodes is above a fraction of the apexes densities. If it is the case, the two overlapping subclasses are merges. If not, the nodes are disconnected therefore creating two clusters. Finally nodes resulting from noise are eliminated from the model. A node is considered as noise if it lies in a low-density area and is weakly connected to other nodes (2 or less edges from/to this node). The fewer neighbours a node has, the higher its density must be in order for it to be kept in the network.

Classification Once the sanitisation has been performed, the network of node is ready to be classified. ESOINN considers the disconnected graphs in the network as being different classes. Therefore, it start by selecting an unlabelled node n, attribute it a class and broadcast this class to all its neighbours, which in turn broadcast the class to their neighbours, until all the node with a path to n are classified. ESOINN repeat this task until every node has a class.

3.2.2 Improvements on the algorithm

However, the ESOINN algorithm is lacking features that are necessary with regard to our approach.

Temporality

The MemorySense project is dealing with time series data for which the time dimension is essential. For this reason, ESOINN, which is by default data agnostic needs a representation of the time. The sanitising task is ran by default every λ input signals. However, if the signals are not regularly distributed over time, this may lead to running the task too often which would cause illegitimate disappearance of nodes from the model or not often enough which would lead to noisy node to perturb the classification. To prevent this behaviour, the λ has been converted to a time value: e.g. $\lambda = 24$ hours. In this way, the sanitising task is not run every k signal but rather every t hours. This measure helps mitigating the difference between active periods when the device records a lot of sensor data and idle periods when the device only records few data points.

Cluster label

Another difference with the original ESOINN regards the way labels are attributed to clusters. Indeed, ESOINN treats each period lambda as a different classification problem and relabels every clusters every time. It is problematic in the case of MemorySense as it can lead to incoherences between two λs: The order of traversal of the graph may differ, leading to different labelling of the same cluster. To address this problem, we introduce a task before the classification routine. In this task, nodes that had already received a label in the last λ period share their class with their neighbours until all the nodes of a connected graph are labeled. However this approach raises issues when clusters are merging or splitting up.

Two clusters can merge when a new edge is created between two graphs that were previously disconnected. This event leads to clusters that contain members from different classes. Two solutions were proposed:

- To apply a majority rule in which, given two merging clusters A and B, the resulting cluster will be classified as A if possesses a total of $points$ higher than B.
- To trigger a user action. In the context of MemorySense, merging clusters imply that two places of interest are becoming one. We can therefore ask the user which of the two places the merged cluster represents.

Another aspect of importance is the splitting of a cluster, which occurs when, after a sanitisation task, an edge of a cluster was removed, leading to two disconnected graph. In this case, the two clusters will have the same class. This behaviour, however, does not represent an

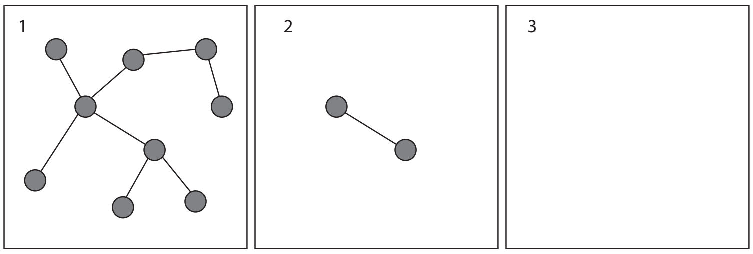


Figure 3.2 – Cluster disappearance after 3 λ

obstacle for MemorySense. Indeed, the two places of interest corresponding to the clusters will continue to exist separately but with the same label. In this way, in case a new edge appears between the two clusters, there will not be a merging conflict. We envisioned however that the user can manually modify the semantic of one of the two PoI resulting from the split, making the two PoI effectively different.

Remembering

A direct consequence of the sanitisation process is that weakly connected clusters will slowly vanish. Removal of low degree nodes combined with the ageing of edges makes it possible for ESOINN to forget nodes progressively. While this mechanism prevents potential mistakes/noise from remaining indefinitely, it also makes legitimate clusters with few corresponding input signals disappear. Indeed, if not enough signals are acquired, the algorithm will not manage to create a network dense enough, therefore allowing the sanitising task to delete these clusters on the long run. We can see in Figure 3.2 that after 3 λ a loosely connected cluster can disappear. This behaviour would be especially detrimental in the context of MemorySense. The activities associated to the PoI in order to generate an semantic episodic memory are entered manually by the user in a bootstrapping phase, and then reused when occurrences of such PoI. Remembering allows retaining this information and therefore avoids to ask redundant information to the user.

Therefore we alter the sanitisation process to prevent clusters reduced to a single node to disappear. More formally, before removing a node with no edge in the network, the algorithm test if the node has a mapping to existing class c, and that no other node in the network maps to c as well. If that is the case, it means that the node was previously determined as part of a cluster and not as noise by the algorithm. The algorithm simply keeps the node in place allowing it to be reactivated at a later time.

Scaling factor

The last modification concerns the insertion of new signals into the network of nodes. The decision of including the new input signal to an existing node or to create a new node is

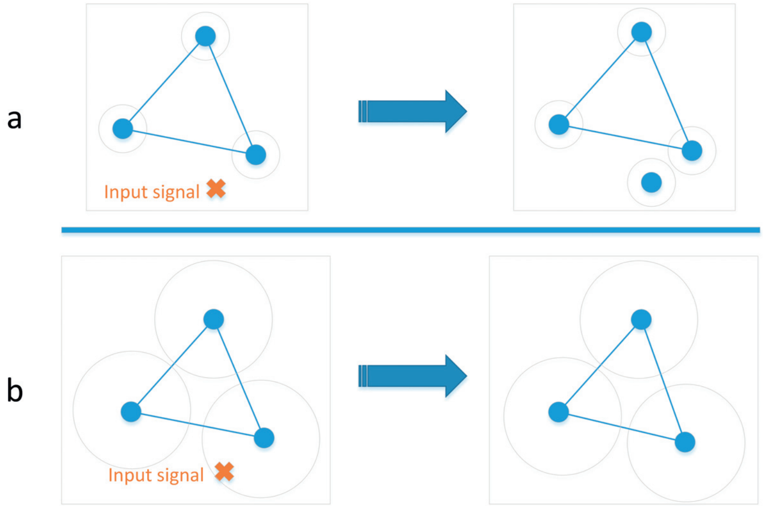


Figure 3.3 – Scaling factor mechanism (a.normal threshold b. scaled up threshold)

based on a threshold determined by the density of the network in the neighbourhood of the input signal. Initial results showed a too large number of nodes as well as too few edges between these nodes. The weak connection of the network leads to nodes being removed in the sanitisation process, as they will be seen as noise.

To tackle this problem, we introduce a multiplicative scaling factor $ThresholdScaling$ into Equation 3.2 to increase the threshold below which a input signal will be attributed to w_1. The influence of this factor is depicted in Figure 3.3. In case a, we can see that with a normal threshold the new input signal would have resulted in the creation of a new node. However, in case b, the input signal falls into the range of one of the nodes. This node will be updated and its edges by reseting their $age = 0$, therefore delaying their potential removal. A direct consequence of this factor is the diminution of the number of nodes in the network, and as a side effect, a speed up in the processing of new input signals.

$$T_n = \begin{cases} \max\limits_{m \in N(n)} (hav(n,m)) * ThresholdScaling & \text{if } n \text{ has any connection in the network} \\ \min\limits_{m \in N} (hav(n,m)) * ThresholdScaling & \text{otherwise} \end{cases} \tag{3.7}$$

The new parameters of the modified ESOINN, applied in the context of MemorySense are provided in Table 3.3. We can see that the λ is taking into account temporality. This implies that the sanitisation process will be run once a day, allowing for new PoI discovered during the day to grow a dense enough network in ESOINN and therefore become a new cluster.

Parameters	Description	Default Value
age_{max}	Age after which an edge will be deleted	100
λ	Period triggering a sanitisation processing	24 hours
c_1, c_2	Node deletion control parameters	0.01, 1.0
$ThresholdScaling$	Factor to increase the original threshold	2.5

Table 3.3 – ESOINN parameters for MemorySense

3.3 Categorisation and representation of SEMs

Once the PoIs detected, it is possible to transform them into SEMs. The temporal dimension ($when$?) of the SEM is covered by the timestamp at which the user entered the PoI. The meaning of the PoI ($where$?) is collected from the Foursquare API [49] which was at the time of the system design the richest semantic dataset for places of interest. Given GPS location, the API allows to retrieve the nearest venues from this location including the name of the venue and its category. When a new PoI is discovered by ESOINN (i.e. a new cluster is generated), MemorySense notifies the user and asks her which of these venues she was visiting. In the future, the venue is attributed to the cluster and is be automatically added in the user's timeline. The activity of the user ($what$?) follows the same approach: Upon detection of a new cluster, the user is prompted to provide a verb and a noun that she can select from the Wordnet database. Once this activity is attributed to the cluster, it will be automatically reused in the next activation of the cluster. the user has however the possibility to edit the activity of a specific SEM at a later time.

We then define two different type of memories which can be categorised as *frequent memories* and *exceptional memories*. Frequent memories are those that occur often in a given time range. They are part of the standard routine of the user, for example staying at home or training in the gym. Exceptional memories on the other hand are those that occur with low frequency and may represent something remarkable in someone's life (e.g., a specific holiday in France, a visit to a new shop). The relative frequency notion that is used here is tightly bound to the specific time range. Episodic elements can be very frequent at the week level (e.g., a week of ski sessions in the Alps), while being exceptional if we consider them at the month or year level. For this reason we keep track of the distribution of the SEM for different levels of granularity: weekly and monthly distributions. We can then provide a ranking of the memories for a specific granularity, whose score is the average score of its underlying episodic elements.

3.4 Evaluation

We evaluate our approach both in terms of performance and feasibility. The two following experiments are based on the GPS trace of users extracted from the dataset of the Nokia challenge organized in 2009 [69] [72]. This dataset contains data from 191 users for a period of over a year. The Nokia time series contains noisy data resulting from the acquisition of

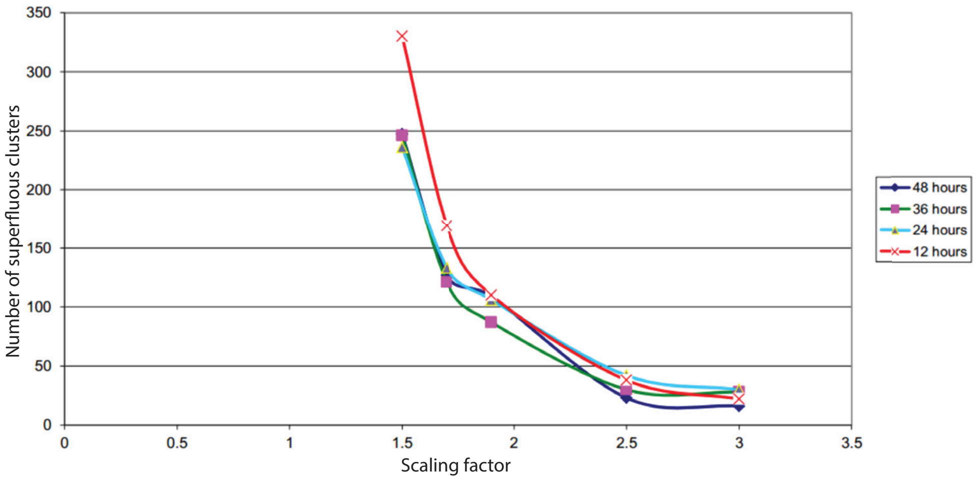


Figure 3.4 – Influence of the scaling factor on the number of superfluous cluster

GPS data-points while the users was traveling. While in our architecture, sensor readings are submitted by the Android activity recognition API which detects if the user is traveling, this piece of information is not available in the Nokia dataset. We therefore consider only the data-points which can be labeled and define the rest as noise which would have been discarded by the Android activity recognition.

3.4.1 Preliminary experiment

An early experiment, focusing on a user for which 133 PoI were available in the ground truth, motivated the implementation of the scaling factor. Preliminary results showed that precision and recall yield excellent results, with a score over 0.999 for both of them. However, this result was biased by the fact that an important number of additional and unwanted cluster are created. Mapping these clusters to the ground truth would lead to a many-to-1 mapping. If we were to push to the extreme this reasoning, we would have a cluster/PoI for each GPS data-point, leading to a perfect precision and recall.

Increasing the scaling factor can effectively generate fewer clusters and avoid the overfitting. This result is depicted in Figure 3.4. The number of extra cluster is lowered down to a dozen of additional clusters.

3.4.2 Clustering experiment

A first evaluation aims at assessing the performance of the recognition of episodic elements. We generate a storyline of PoI, comparing our modified version of the ESOINN algorithm against the clustering algorithm DBSCAN. This algorithm was used by Montoliu et al. in [85] to detect places of interest. Since DBSCAN is also a density-based algorithm, it shares similar features to ESOINN. However, DBSCAN has a full knowledge of the dataset when detecting PoIs, it is therefore interesting to compare it to our approach which only have an incremental vision of the dataset.

We begin the evaluation by generating 3 alternate storylines for each algorithm:

- one containing all the PoIs that the user visited
- one containing only the PoIs which are considered exceptional at the scale of a month. We consider an event to be exceptional if it represents less than 1.5% of events of the current month.
- another storyline contains only frequent events. Events are considered frequent if they account for at least 10% of events of the current month.

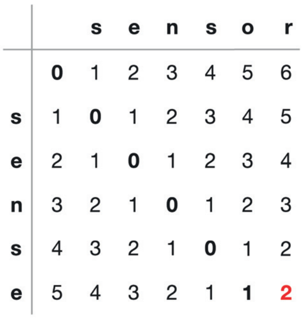
In order to assess the performance of our approach, we need to compare the places of interests discovered by MemorySense compared to the ground truth provided by the Nokia dataset. This ground truth is given as regions where the user stayed more than 10 minutes.

We perform a mapping between the discovered place of interest represented by the clusters and the ground truth, by mapping each clusters with the majority of ground truth found among the data-points that belong to this cluster. For instance if a cluster C contains 90% of data-points that are labeled as belonging to $PoI-1$, 5% labeled as $PoI-2$ and 5% labeled as $PoI-3$, the cluster C will be mapped to $PoI-1$. From this mapping, we can deduce a measure of precision and recall.

Once the 6 timelines {exceptional events , frequent events, all events} x {modified ESOINN, DBSCAN} are generated, they need to be compared with the timeline of the ground truth. To this purpose we use an approach similar to the Levenshtein distance, also known as edit distance. This distance computes the minimum number of steps necessary to transform a list of object L_1 into another list L_2. It allow the following steps:

- $Insertion$ Add an element to L_1 (cost 1)
- $Substitution$ Replace an element of L_1 by an element of L_2 (cost 1)
- $Deletion$ Remove an element from L_1 (cost 1)
- $(Match)$ Elements are the same for both lists at a specific position (cost 0)

For example, considering two lists or character $L_1 = [s, e, n, s, e]$ and $L_2 = [s, e, n, s, o, r]$, the edit distance between the two lists is $editDistance(L_1, L_2) = 2$ as described in Figure 3.5. Indeed, one would need to substitute the last e of L_1 into an o, and append a r to transform L_1 into L_2. More valuable than the distance itself, the Wagner–Fischer algorithm that is standardly used to compute the Levenshtein distance allows to establish the minimal series of steps that must



		s	e	n	s	o	r
	0	1	2	3	4	5	6
s	1	**0**	1	2	3	4	5
e	2	1	**0**	1	2	3	4
n	3	2	1	**0**	1	2	3
s	4	3	2	1	**0**	1	2
e	5	4	3	2	1	**1**	**2**

Figure 3.5 – Edit distance between two words

be used to convert L_1 into L_2. In Figure 3.5, the algorithm starts from the top left element of the matrix and iteratively figures the less costly path (in bold) that leads to the bottom right element of the matrix. The series can be read as [*match, match, match, match, substitution, insertion*], with respective cost of $[0,0,0,0,1,1]$.

In this evaluation we define the storyline as lists of PoI and we determine what is the minimum series of steps needed to transform the evaluated storyline to the ground truth. PoI belonging to the original storyline may not appear in the storylines produced by the algorithms ($Deletion$), and PoI not present in the true storyline may be inserted ($Insertion$) in the produced storyline. This issue is handled by introducing a $null$ class in the confusion matrix. We remark that the precision and recall of the $null$ class is not taken into account in the computation of the overall precision and recall, and only false negatives (i.e., an element is deleted) and false positives (i.e., an element is created) introduced by the $null$ class are factored in the performance metric. In this way, the bias introduced by the $null$ element can only impact negatively the precision and recall. The precision and recall values of the different algorithms for each storylines of 4 users of the Nokia dataset with the longest GPS traces are presented in table 3.4.

While the results for the storyline containing all events are similar for both algorithms, the two alternate storylines display different performances regarding the algorithm. Our modified ESOINN performs better on storyline of frequent events compared to exceptional events. This is due to the algorithm requiring enough input signals at the same location before having a stable cluster. However, we can observe that our approach is globally comparable to using DBSCAN in terms of precision and recall and a second experiment related to the running time of the algorithms should further justify our choices.

3.4.3 Performance evaluation

To evaluate the applicability of our approach in a mobile environment, the two algorithms DBSCAN and modified ESOINN are ran on an Android smartphone with a CPU of 1.5 GHz quadcore, 2GB of RAM and the operating system Android 4.4. We processed the GPS trace

	Modified ESOINN		DBSCAN	
storyline	precision	recall	precision	recall
every events	0.92	0.89	0.92	0.86
exceptional events	0.71	0.61	0.95	0.77
frequent events	0.93	0.93	0.83	0.83

Table 3.4 – Average precision and recall of the location clustering

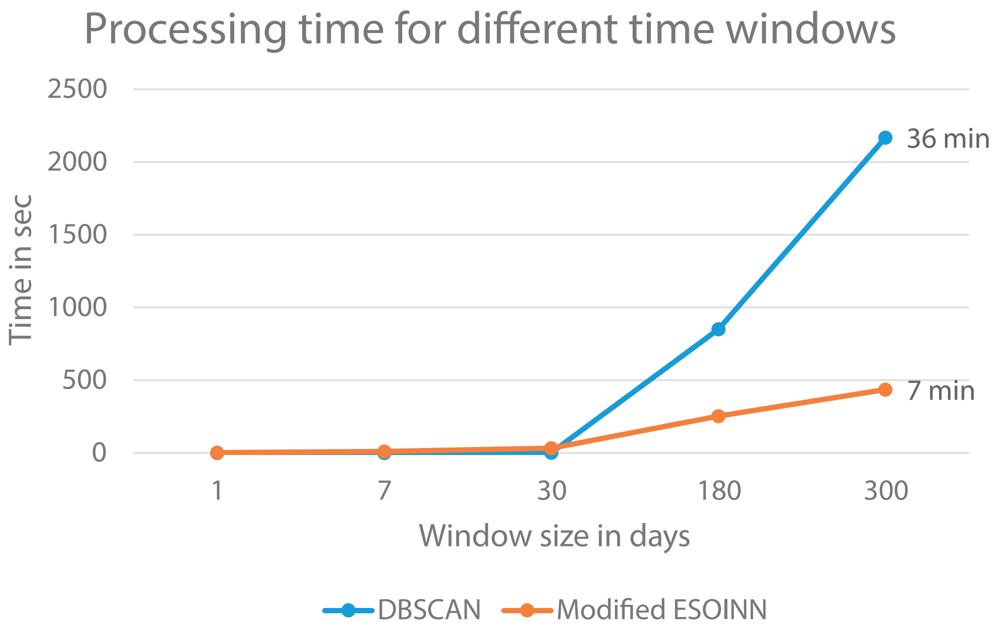


Figure 3.6 – Mobile performance of the modified ESOINN

containing 13674 GPS points in the format <*timestamp, latitude, longitude*>, for which a ground truth is provided. As depicted in Figure 3.6, the processing time for DBSCAN increases dramatically with the length of the trace. Since DBSCAN is not incremental, after 180 days (≃ 6 months), DBSCAN would need to process all the 179 previous days to generate the daily storyline, with a processing time of 14 minutes and 12 seconds, while ESOINN would need on average 2.60 seconds. Although we envision this daily processing to happen when the smartphone is charging, in order to limit the impact on the user experience, the growth of DBSCAN processing time will eventually become unpractical and exceed the charging time itself. This experiment justifies the use of ESOINN in a mobile environment.

3.5 Conclusion

In this chapter we presented a mobile approach to generate episodic elements based on GPS sensor. We motivated the use of the ESOINN clustering algorithm presented by Furao for PoI detection, by explaining the features that are necessary to address such challenge.

These features are both in term of clustering properties (unsupervised, unbounded number of cluster, density-based) as well as energy efficiency (online, incremental). We then presented the enhancements performed and the algorithm required in order to properly handle time series of GPS locations. These improvements include adding a notion of temporality to the algorithm, maintaining consistent cluster labels, processing cycles, preventing loss of clusters and adjusting the density scaling. Our evaluation shows that ESOINN is suitable for PoI detection in term of clustering capabilities, and that it can be used in a mobile environment without impact on the performances.

Following the hierarchical framework of this thesis, aligned with the episodic memory model of Conway, the generation of SEMs is a stepping stone towards building more complex entities such as CEMs. We present in the next chapter how these memories can be combined in order to build complex episodic memories which abstract and semantically enhance simple episodic memories.

4 Routine recognition

4.1 Introduction

The ever-increasing sensing and computing capabilities of modern smartphones are enabling new unobtrusive ways to collect the digital trails of users, organise them and extract meaningful information regarding their activities. Thanks to sensors such as GPS, accelerometer and microphone, several user-centric smartphones applications have been recently developed to sense and understand the way we live and the world that surrounds us [79]. These applications typically focus on recognising specific aspects of the user life and creating long sequences of user-related states, such as the physical activity (e.g., walking, standing, running, etc.) or the visited locations (e.g., home, workplace, etc.). However, how to identify complex patterns, or routines, from sequences of simple user states is still an ongoing challenge.

In this chapter, we introduce two incremental algorithms for fusing simple episodic elements (SEMs) into complex episodic memories (CEM) that we characterise as complex routines, following an established human episodic memory model. Applications that could benefit from such approach range from elder care with the monitoring of patients' routines, to personalised recommendations (e.g., fitness advices, activity suggestions, reminders, etc.). We evaluate our framework using a comprehensive data set containing rich geographic information. The evaluation is both in terms of computational requirements of the routine reconstruction and accuracy of the segmentation of the sequence of states. Furthermore, we also use a state-of-the-art algorithm, T-pattern, to assess the accuracy of our approach.

The contributions of this chapter are two models based respectively on frequent pattern mining and on finite state machines, to generate, in an unsupervised way, sets of CEMs (each of them corresponding to complex patterns) from sequences of SEMs.

The organisation of the rest of the chapter is as follow: In Section 4.2, we briefly introduce how the simple episodic memories are generated. In Section 4.3, we describe the two approaches taken to generate routines. In Section 4.4, we evaluate these two approaches against a real life dataset and compare their accuracy again the T-pattern algorithm. We conclude with a summary in Section 4.5.

4.2 Generation of simple episodic memories

The system architecture of MemorySense in Figure 2.2 describes the two different sources for SEMs. They can come from virtual sensors (e.g. giving a phone call, receiving a message) or from GPS sensors via the detection of PoIs introduced in Chapter 3. The places of interest are augmented with semantics information that is queried from external services. We extract the name of the place (e.g. Pizza Fratel) as well as the type of the place (e.g. restaurant) and in the implementation, the activity performed by the user is selected from a list of activity the user already performed at this type of place (e.g. eat pizza).

4.3 Generation of complex episodic memories

Once the SEMs are generated, they can be aggregated, according to the episodic memory model, into complex episodic memories or routine (In the remaining of these chapter we will use the two concepts interchangeably). When a new CEM is created, the user can provide MemorySense with a label (i.e. Picnic with family and friends) which will be reused when posterior instances of the CEM are detected. In the system presented in this chapter, we fix the granularity of CEMs by assuming that they have the timespan of a day. This assumption simplifies the research problem at hand while keeping it realistic: weddings, week days, excursions can be seen as an aggregation of multiple SEMs over a day. The generalisation of CEM of variable length is left as future work.

The main challenge associated with the routine detection concerns the management of the semantic noise. As explained in the introduction of this thesis, while simple events such as SEM need to address the noise at the sensor level, more complex events such as routines are impacted by another level of noise that we call semantic noise. In the specific case of routine, this noise is characterised by small variations of the routine. We consider three variations:

- Addition of SEM: in this variation, a routine is altered by having an exceptional SEM inserted (e.g. passing by the post office exceptionally before going to work)
- Deletion of SEM: This variation removes an SEM that is usually present from the routine. (e.g. skipping lunch)
- Reordering SEM: This variation requires the notion of sequentiality and parallelism in routine. Some SEM can be present in a routine always in the same order (e.g. Waiting at the metro stop and then working at the office). The fact that these elements are always in the same order offers additional information to detect such routine. Conversely, some SEMs can be experienced in indifferent order, such as eating an ice cream before or after going to the movie. In some cases the ordering of these SEMs can be reversed while

having the same routine.

- Substitution: This variation simply consist in replacing an SEM by another, similar or not (e.g. enjoying a spa instead of going to the cinema). Substitutions are harder to leverage to detect routine.

4.3.1 Finite state machine model

Finite state machine (FSM) is a mathematical model designed to represent different states and the possible transitions between these states. Such models have been used in the literature in a wide range of contexts from software design testing [28] to recognition of activities [116] and gestures [65]. In the context of memorysense, the first step towards the creation of complex episodic memories using the FSM model is to create sequences of SEMs, corresponding to one day of the life of the user. To do that, we discretise the time dimension into k buckets. For each bucket, there is a single SEM, corresponding to the most representative activity occurred in the time bucket. In case that more than one SEM have been generated for that bucket, we select the most representative one with the majority rule. Therefore, at the end of one day in the user life, a sequence $\{sem_1, sem_2, \ldots, sem_k\}$ is generated by the system. Each SEM sem_j is defined as $< \ell, t >$, where $\ell \in \mathscr{L}$ is the location type, $\mathscr{L}$ is the set of location types, and $t \in \mathscr{T} = \{1, \ldots, k\}$ is the time bucket.

A complex episodic memory is modelled as a finite state machine $< \mathscr{S}, s_0, \Sigma, \Lambda, T >$, where $\mathscr{S}$ is the set of states, s_0 is the initial state, Σ is the input alphabet, Λ is the output alphabet, and $T : \mathscr{S} \times \Sigma \rightarrow \mathscr{S} \times \Lambda$ is the transition/output function. In our modelling, $\mathscr{S}$ is the set of all possible location type-time pairs, $\mathscr{S} \subseteq \mathscr{L} \times \mathscr{T}$, the input alphabet Σ is the set of location type $\mathscr{L}$, and the output alphabet Λ is the set of integer values $\mathbb{Z}$, representing the number of times a specific location type have been observed when transiting from one state to another. Figure 4.1 shows an example of FSM, corresponding to a typical working day, where the time has been discretised in 5 buckets.

For each user, a set of FSM are incrementally created in an unsupervised way from the sequences of SEM that are generated at the end of each day. Given a sequence of SEM $\{sem_1, sem_2, \ldots, sem_k\}$, the system has to decide (a) whether the sequence is an instance of an existing FSM, possibly modifying the FSM to accommodate the sequence of SEM or (b) to create a new FSM, since the given sequence is not well matched by any existing FSM. This decision is described in the algorithm 1 and can be explained as follows:

1. Given a candidate FSM, we temporarily update it with the sequence of SEMs, possibly adding new states, increasing the output value on each transition between existing states or creating new transitions with output value 1 if such transitions do not appear in the original FSM.

2. We compute the degree of match DoM between the sequence of SEMs and the FSM as

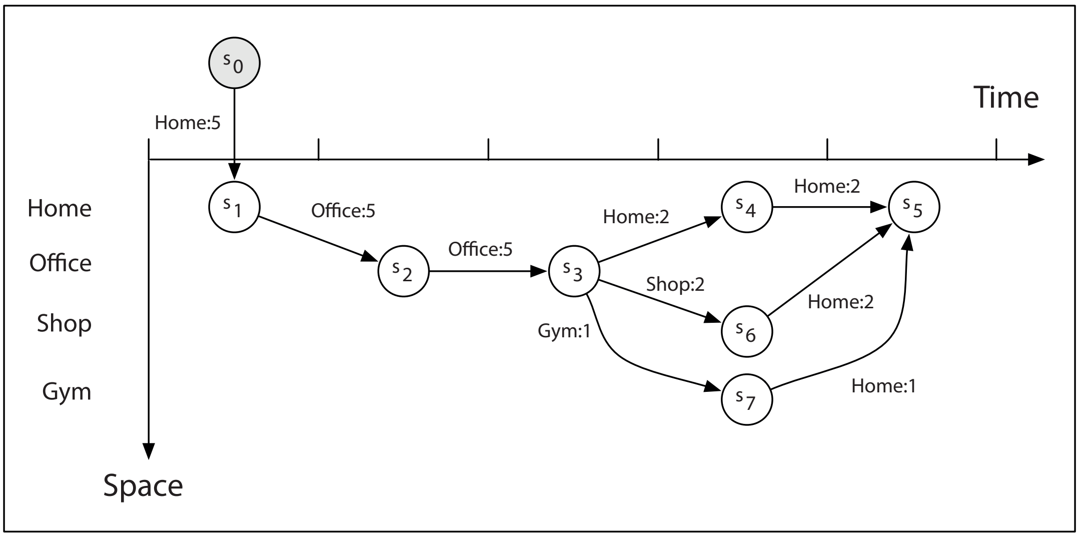


Figure 4.1 – Example of finite state machine.

the product of each output value of the state transitions triggered by the sequence of SEMs, normalised by N_{FSM}, that is, the number of sequences of SEMs that have been matched with the FSM under evaluation.

3. If the degree of match is below a certain threshold θ, it means that the given FSM is not a good match for the sequence of SEMs, otherwise the FSM is marked as possible candidate.
4. If the list of candidate FSM is empty, a new FSM is created, corresponding to the sequence of SEMs.
5. If the list of candidate FSM is not empty, the sequence of SEMs is matched with the FSM with the highest DoM. For all the other discarded FSMs, the temporary modifications to the states, transitions and output values are undone.

The threshold θ of the degree of match with a given FSM is defined as $\frac{1}{N_{\text{FSM}}}^{\lambda}$, where N_{FSM} is the number of sequences of SEMs that have been matched with the given FSM, and $\lambda \in \{1, \ldots, k\}$ is a parameter that control how much diversity between the sequence of SEMs and the FSM is accepted to consider the FSM a good match for the sequence of SEMs.

Figure 4.2 shows the tentative match between the sequence of SEMs {Home, Office, Bar, Cinema, Home} and the existing FSM of Figure 4.1. The output value of the transitions $s_0 \rightarrow s_1$ and $s_1 \rightarrow s_2$ is increased by 1, and the states and transitions corresponding to Office→Bar, Bar→Cinema and Cinema→Home are added to the FSM with output value 1. Assuming that $N_{\text{FSM}} = 6$, the degree of match in this case is $\frac{6}{6} \cdot \frac{6}{6} \cdot \frac{1}{6} \cdot \frac{1}{6} \cdot \frac{1}{6} = 0.0046$. If the threshold $\theta = \frac{1}{6}^{\lambda}$ is parameterised with $\lambda \in \{1, 2, 3, 4, 5\}$, the degree of match between the sequence of SEMs and the FSM is above θ, thus putting the FSM into the candidate set.

Algorithm 1 Finite State Machine Matching

```
Input: models, SEMs
Output: selectedFSM
  degrees = Map(FSM, degree)
  for FSM in models do
      UPDATE(FSM, SEMs)
      degrees.add COMPUTEDEGREE(FSM, SEMs)
      UNDO(FSM, SEMs)
  if selectedFSM != null then
      UPDATE(selectedSEM, SEMs) return selectedFSM
  else
      selectedFSM = generateNewFSM(SEMs)

  function UPDATE(FSM, SEMs)
      previousEE = FSM.startEvent
      for SEM in SEMs do
          if FSM.contains(SEM) then
              FSM.get(SEM).incrementCounter
          else
              FSM.add(SEM)
          if FSM.containsDirectedEdge(previousSEM, SEM) then
              FSM.incrementEdge previousSEM, SEM
          else
              FSM.createEdge previousSEM, SEM
          previousEE = EE

  function UNDO(FSM, SEMs)
      previousSEM = FSM.startEvent
      for SEM in SEMs do
          FSM.get(EE).decrementCounter
          FSM.decrementEdge previousSEM, SEM
          previousSEM = SEM

  function COMPUTEDEGREE(FSM, SEMs)
      weight = 1
      previousSEM = FSM.startEvent
      for SEM in SEMs do
          weight = weight * getWeight(previousSEM, SEM)
          previousSEM = SEM
      return weight/FSM.numberOfActivations
```

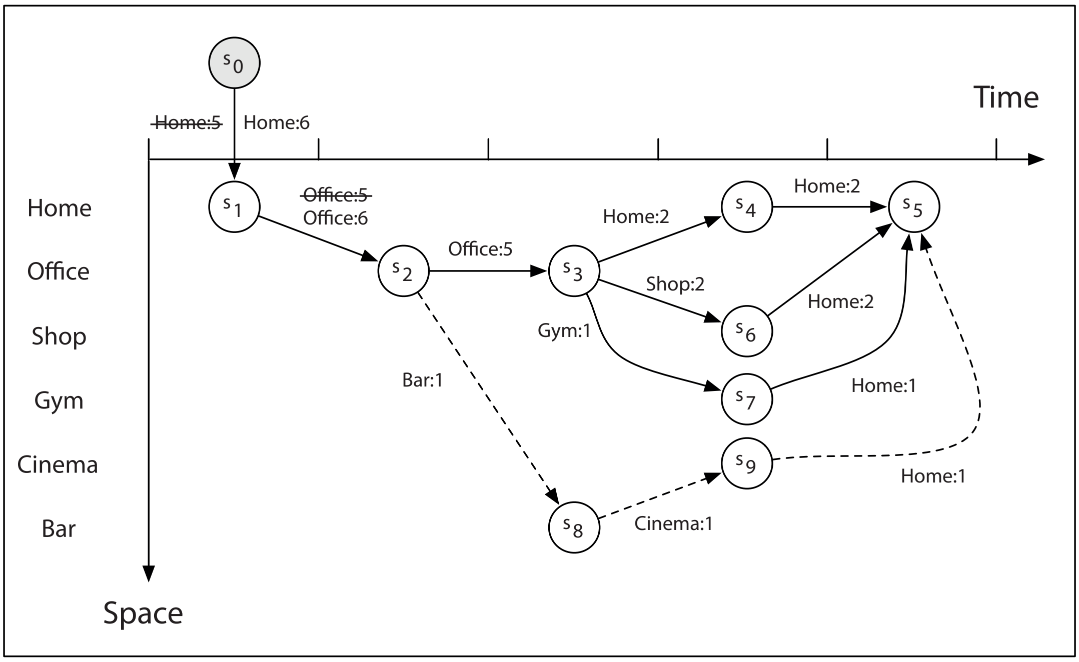


Figure 4.2 – Example of matching of a sequence of SEMs against an existing finite state machine.

By attributing sequences of SEM to existing state machines or by generating new ones, we propose an incremental method to effectively aggregate SEMs into CEMs, each CEM being modelled by the corresponding finite state machine. The proposed model is flexible regarding the insertion of additional SEMs, the deletion of SEMs or the substitution of one SEM by another. However, the intrinsic time-ordering of the buckets makes it very order-dependent and not suitable for unordered SEMs.

4.3.2 Frequent pattern model

Approach overview

We propose an alternative approach for the generation of CEM based on incremental frequent itemset mining and frequent sequence mining. The concept of frequent itemset was popularised by Agrawal et al. [3] (under the name *large* itemset) as an essential component for detecting association rules from large databases of sets of items as for example supermarket baskets database, or web logs [84]. Frequent itemset mining aims at finding in a list of transactions, the (unordered) frequent sets of items which are present in more than s_i transactions, where s_i is called the minimum support necessary to be defined as frequent itemset. Conversely, sequence mining aims at finding the (ordered) frequent sequences which are present in more than s_s transactions. These methods offer the advantage of working with transactions of different sizes, making the bucketing introduced in the previous model unnecessary. A second advantage is that the combination of the two methods allow for the detection of both ordered and unordered patterns, making this approach more flexible than the finite state machines. Itemsets and sequences being two closely related concepts, we will, when possible, use the term $pattern$ to refer indifferently to one or the other.

Once frequent patterns of both types have been detected, a list of boolean features is computed for each CEM based on whether the CEM matches a specific pattern or not. Finally a hierarchical clustering is performed on these binary features in order to group CEM which share similar patterns.

Incremental frequent itemset mining

The frequent itemset mining algorithm used in MemorySense is based on the FUP algorithm [25]. Although more recent and more efficient algorithms have been developed, FUP has the advantage of being base on the Apriori generation of itemsets which can also be altered for sequence mining. It is therefore possible to share the required scans of the database for both frequent itemset mining and sequence mining.

Apriori algorithm The Apriori algorithm, introduced by Agrawal in [4] proposes a bottom up approach at building frequent itemsets. More formally, it considers a database $\mathscr{DB}$ of transactions defined as a set of items: $T = \{i_1, i_2\}$ and a minimum threshold s_i .

The algorithm is initialised by scanning $\mathscr{DB}$ and building $\mathscr{L}_1$ the set of frequent 1-itemsets such that $\forall X \in \mathscr{L}_1, support(X, \mathscr{DB}) > s_i$, where $support(X, \mathscr{DB})$ represent the number of transactions T for which X is a subset. In the later stages, the algorithm generates larger k-itemsets for $k = 2, 3...$ and then scan $\mathscr{DB}$ to only retain the frequent k-itemsets. The algorithm ends when it reaches a k for which none of the k-itemsets are frequent.

The interest of the Apriori algorithm lies in the generation of $\mathscr{L}_k$. Instead of computing all possible the candidate k-itemsets and then scan $\mathscr{DB}$ to retain only the frequent ones, the algorithm relies on the $\mathscr{L}_{k-1}$ frequent $(k-1)$-itemsets and the lemma that **all subset of a large itemset must also be large** which proof is provided in [4] . Therefore, candidate (k)-itemsets can be built by taking the union of elements in $\mathscr{L}_{k-1}$ that differ by only one item. This technique prevents an otherwise exponentially growing number of candidate itemsets and is presented in Algorithm 3.

FUP algorithm The FUP algorithm is designed to incrementally maintain existing frequent itemsets and discover new ones in an optimised fashion when the database $\mathscr{DB}$ of size D evolves in time by appending increments of data db of size d. The key idea of the algorithm is to limit the number of candidate itemsets that are generated by performing initial computation on db which is typically orders of magnitude smaller than $\mathscr{DB}$.

Similarly to the Apriori algorithm, FUP is presented in two similar phases: 1/ the processing of the 1-itemsets and 2/ the processing of bigger k-itemsets. Let $\mathscr{L}_k$ be the frequent k-itemsets before the increment update, $\mathscr{L}'_k$ the updated frequent k-itemsets, s the relative support threshold for an itemset to be frequent. The main steps of the algorithms are the following:

1. Scan db and update the support of all itemsets $X \in \mathscr{L}_1$. After the scan, the itemsets X which verify $X.support < s * (D + d)$ are, by definition, not frequent anymore and can be discarded. The remaining itemsets are added to $\mathscr{L}'_1$.
2. In the same scan of db, a candidate set $\mathscr{C}_1$ stores the 1-itemsets that are generated from the items found in db and are not already present in $\mathscr{L}_1$. After the scan of db, $\mathscr{C}_1$ is pruned by removing the itemsets X for which $X.support < s * d$. Indeed, if X was not a frequent itemset previously in $\mathscr{DB}$, it cannot be frequent after the update.
3. Scan the original database $\mathscr{DB}$ to determine the support of the remaining candidate itemsets in $\mathscr{C}_1$. After the scan, select the itemsets X which verify $X.support > s * (D + d)$ and include them in $\mathscr{L}'_1$.

At this point of the algorithm, $\mathscr{L}'_1$ contains the frequent 1-itemsets of the updated database. The remaining of the algorithm builds sets of frequent k-itemsets of increasing sizes iteratively.

4. This steps focuses on identifying the itemsets in $\mathscr{L}_k$ that are no longer frequent after the update. This is done in two ways: a/ Using the lemma introduced in the Apriori algorithm, an itemset $X \in \mathscr{L}_k$ cannot be frequent if it contains subsets that were discovered as non frequent in the previous steps. Therefore, any set $X \in \mathscr{L}_k$ which has a subset Y such that $Y \in \mathscr{L}_{k-1} - \mathscr{L}'_{k-1}$ cannot be frequent and can be discarded. b/ Similarly to step 1, the algorithm scans db and updates the support of all itemsets $X \in \mathscr{L}_k$. After the scan, the itemsets X which verify $X.support < s * (D + d)$ are, not frequent anymore and can be discarded. The remaining itemsets are added to $\mathscr{L}'_k$
5. Generate a candidate set $\mathscr{C}_k$ by applying the generation algorithm of Apriori (Algorithm 3) on $\mathscr{L}'_{k-1}$ and remove from $\mathscr{C}_k$ the k-itemsets present in $\mathscr{L}_k$ (these itemsets have already been tested in the previous step). Similarly to step 2, db is scanned to compute the support of the candidates $X \in \mathscr{C}_k$. After the scan, $\mathscr{C}_k$ is pruned by removing the itemsets X for which $X.support < s * d$ (following the same reasoning that step 2).
6. At this step, the algorithm have reduced the number of candidates $\mathscr{C}_k$ as much as possible without having scanned $\mathscr{DB}$. Similarly to step 3, the algorithm then scans the original database $\mathscr{DB}$ to determine the support of the remaining candidate itemsets in $\mathscr{C}_k$. After the scan, it selects the itemsets X in $\mathscr{C}_k$ which verify $X.support > s * (D + d)$ and include them in $\mathscr{L}'_k$.

The steps 4 to 6 are repeated for $k = 2, 3, \ldots$ until it reaches a k for which $\mathscr{L}'_k = \emptyset$.

Modified FUP algorithm FUP, as most of the incremental itemset mining algorithm uses a minimum support s_i relative to the size of the database (e.g. $s_i = 10\%$ means that to become frequent, the itemset needs to appear in at least 10% of the transactions present in the database). This is not suitable for MemorySense as it would allow itemsets that were previously frequent to become non-frequent if they don't appear in the next increments. This behavior would be equivalent to forgetting. Although forgetting is a property of the brain, it is not suitable in our case to forget. Information which may have required some user input can be deleted and, at a later time, the user would be prompted again for the same information because the itemset would have become frequent again. To this purpose, FUP is adapted for absolute support(e.g. $s_i = 7$ means that to become frequent, the itemset needs to appear in at least 7 transactions present in the database).

The absolute minimum support imposes a several modifications on the original FUP algorithm. Indeed, if a k-itemset is already frequent in $\mathscr{DB}$ before an increment, having the present or not of the itemset in the increment db will not have any impact, the itemset will still be frequent. Similarly, if a k-itemset is already found frequent in the increment db, it can directly be added to the list of updated frequent itemsets $\mathscr{L}'_k$. A negative aspect of having an absolute minimum support is that the number of itemself is always increasing as more increments of data are added. However, as we will see in the evaluation section, in the case of MemorySense, this number does not impact the performance of the algorithm.

- Step 1 of the original algorithm is not useful anymore since, once a itemset became frequent, it will remain frequent
- In step 2, after scanning db k-itemset with support over the min support can directly be added to $\mathscr{L}_k$
- Step 3 remains the same
- In step 4, we benefit from the Apriori lemma to prune from $\mathscr{L}_k$ the itemsets which have subsets Y such that $Y \in \mathscr{L}_{k-1} - \mathscr{L}'_{k-1}$
- In step 5, the generation of the candidate itemsets $\mathscr{C}_k$ is not change compared to the original FUP algorithm. However, after scanning the db increment, the k-itemset with support over the min support can directly be added to $\mathscr{L}_k$
- Step 6 remains the same

We define a database $\mathscr{DB} : [day_1, day_2, ..., day_n]$, where $day_i = [sem_{i1}, sem_{i2}, ..., sem_{id}]$ represents a day i as an ordered sequence of d SEMs and an increment $increment : [day_{n+1}, day_{n+2}, ..., day_m]$. Following the same notation that in the previous paragraph, we present the modified FUP algorithm in Algorithm 2. It is used to process new increments and update the list of frequent itemsets. It follows the bottom-up approach of the standard Apriori algorithm, but makes use of the list of frequent itemsets discovered in previous iteration to reduce the necessary computation performed at each scan.

The modified FUP algorithm will generate candidate itemsets of size $k+1$ using the method presented in algorithm 3. This method makes use of the frequent itemsets of size k and is base on the same lemma than the original Apriori algorithm: a k-itemset can be frequent only if all of its subsets are frequent. The modified FUP algorithm will then test these candidates against the database and retain only the frequent candidates. This process is repeated recursively until the set of retained frequent candidates is empty.

Incremental frequent sequence mining

In parallel to frequent itemset mining, we use frequent sequence mining to be able to capture the temporal relationships between SEMs when available. To this purpose we use the GSP algorithm [113]. This algorithm, originally not incremental, modifies the generation of the candidate itemsets of the Apriori algorithm to enforce the time-ordering of the sequence. For example $\mathscr{L}_{k-1} = \{A \rightarrow B, A \rightarrow C\}$ will lead to the generation of candidate sequences $\{A \rightarrow B \rightarrow C, A \rightarrow C \rightarrow B, A \rightarrow BC\}$. We can note that the right hand side of the last sequence is composed of simultaneous event B and C which is assumed impossible in MemorySense: the user is doing only one activity at a time. The original GSP algorithm contains additional features such a definition of a minimum or maximum gap between elements of a sequence or definition of a sliding window within which the elements must be found in order to be eligible for becoming a frequent sequence. We do not need these features as they were initially intended to filter shopping basket databases, which is why these features are not retained in the context of MemorySense and therefore, not implemented. Our algorithm for sequence mining is similar to the modified FUP algorithm (algorithm 2) presented previously, the difference being that the generation of the candidate sequences is done using the sequence candidates generation of GSP as defined in algorithm 4.

The choice of the FUP algorithm is therefore motivated by the fact that both frequent itemsets and frequent sequences can be discovered during the same iteration of FUP. This property allow for less scan of the database further reducing the computational cost of our approach.

Generation of binary features

Definition. A pattern (itemset or sequence) p is closed if there exist no superset s such that $support(p) = support(s)$.

Once frequent itemsets and sequences are discovered, each day can be attributed a list of boolean features: for each itemset and sequences denoted as patterns, if the day matches the pattern, the corresponding boolean is set to true, false otherwise. At this stage, frequent itemsets and sequences are filtered in order to retain only meaningful patterns. To this

Algorithm 2 Frequent itemset mining

```
Input: $\mathscr{DB}, db, \mathscr{L}_k, s_i$ // absolute support
Output: $\mathscr{L}'_k$ //updated
  // Initial k = 1
  for day in $db$ do
      SEM = getUniqueSEMs(day)
      if $X \in \mathscr{L}_1 in \{SEM\}$ then
          X.support++
      else
          $\mathscr{C}_1 = \mathscr{C}_1 \cup X$
  for $X \in \mathscr{C}_1$ do
      if $X.support > s_i$ then
          $\mathscr{L}'_1 = \mathscr{L}'_1 \cup X$
          $\mathscr{C}_1 = \mathscr{C}_1 - X$
  for day in $\mathscr{DB}$ do
      SEM = getUniqueSEMs(day)
      if $X \in \mathscr{C}_1 \subset \{SEM\}$ then
          X.support++
  for $X \in \mathscr{C}_1$ do
      if $X.support > s_i$ then
          $\mathscr{L}'_1 = \mathscr{L}'_1 \cup X$
    //for k> 1
  if then$|\mathscr{L}_{k-1}| > 1$ update $\mathscr{L}'_k$:
      $\mathscr{C}_k = \emptyset$
      $\mathscr{L}'_k = \emptyset$
      for day in $db$ do
          SEM = getUniqueSEMs(day)
          if $X \in \mathscr{L}_k \subset \{SEM\}$ then
              X.support++
              $\mathscr{L}'_k = \mathscr{L}'_k \cup X$
      $\mathscr{C}_k$ = aprioriGen($\mathscr{L}_{k-1}$)
      $\mathscr{C}_k = \mathscr{C}_k - \mathscr{L}'_k$
      for day in $\mathscr{DB}$ do
          SEM = getUniqueSEMs(day)
          if $X \in \mathscr{C}_k \subset \{SEM\}$ then
              X.support++
      for $X \in \mathscr{C}_1$ do
          if $X.support > s_i$ then
              $\mathscr{L}'_k = \mathscr{L}'_k \cup X$
```

Algorithm 3 Apriori generation

```
Input: $\mathscr{L}_{k-1}$
Output: $\mathscr{L}_k$
  sort($X \in \mathscr{L}_{k-1}$)
  $\mathscr{L}_k = \{\}$
  for $X \in \mathscr{L}_{k-1}$ do
      for $Y \in \mathscr{L}_{k-1}$ do
          if $(X - X.last) == (Y - Y.last)$ then
              $\mathscr{L}_k = \mathscr{L}_k \cup \{X \cup Y\}$
  return $\mathscr{L}_k$
```

Algorithm 4 Sequence Apriori generation

```
Input: $\mathscr{L}_{k-1}$
Output: $\mathscr{L}_k$
  sort($X \in \mathscr{L}_{k-1}$)
  $\mathscr{L}_k = \{\}$
  for $X \in \mathscr{L}_{k-1}$ do
      for $Y \in \mathscr{L}_{k-1}$ do
          if $(X - X.last) == (Y - Y.last)$ then
              $\mathscr{L}_k = \mathscr{L}_k \cup \{(X - X.last) \to Y.last \to X.last\}$
              $\mathscr{L}_k = \mathscr{L}_k \cup \{(X - X.last) \to X.last \to Y.last\}$
  return $\mathscr{L}_k$
```

purpose patterns containing a single element are filtered out. Indeed, they just inform about the presence of an element without adding extra information. As an additional filtering, only closed frequent patterns are retained. This is to avoid duplication of information.

Indeed, following the definition, if a pattern is not closed, it contains the same amount of information as its superset with equal support. The consequence for our list of boolean features is that for each day, the boolean corresponding to the open pattern and the boolean corresponding to its superset with equal support would always have the same value, hence the redundancy in information.

Hierarchical clustering

Once the boolean features list has been defined for each day, they can be clustered based on this list. Indeed, days having a high number of patterns in common should be classified as belonging to the same CEM. To this purpose we use an agglomerative hierarchical clustering approach using a maximum distance between clusters as a stopping criterion.

The distance between 2 days is taken as the hamming distance between their respective boolean features. The hamming distance can be defined as the number of positions at which the element of two ordered list differ (e.g. $Hamming(01\mathbf{00}1, 01\mathbf{11}1) = 2$).

The linkage (i.e. distance between two clusters) is taken as full linkage to avoid any "chaining effect". Therefore the linkage of two clusters $A = \{cem_1, cem_2, ..., cem_k\}$ and B=$\{cem_1$, cem_2, ..., $cem_j\}$ is defined as $max\{a \in A, b \in B, Hamming(a, b)\}$.

We can notice that the clustering is not done incrementally. However, the growth of the number of frequent patterns is much slower than the growth of the database and therefore the complexity added by an incremental algorithm would not be justified.

Example

In this example we cover the entire pipeline of generation of CEMs using frequent pattern mining. The days of the user and their associated SEMs are described in table 4.1. We set the minimum support threshold of the frequent itemset mining (S_i) and the minimum support threshold of the frequent sequence mining (S_s) as $s_i = s_s = 3$. The stopping criterion of the hierarchical clustering is set to $maxDistance = 1$.

Day	SEMs
1	home, work, restaurant, work, home
2	home, work, restaurant, work, supermarket, home
3	home, work, sport center, work, parents in law
4	home, supermarket, sport center, restaurant, hotel

Table 4.1 – Dataset example

The frequent itemset mining result is given in table 4.2. As we can see, for an itemset size k, only the frequent $(k-1)$-itemsets are used to generate the candidates.

itemset size	{itemsets}	support
1	$\{home\}$	4
	$\{work\}$	3
	$\{restaurant\}$	3
2	$\{home, work\}$	3
	$\{home, restaurant\}$	3

Table 4.2 – Frequent itemsets

The frequent sequence mining result is given in table 4.3. The results are very similar to the frequent itemsets for two reasons: First, the same minimum support threshold was used for both pattern mining techniques. Secondly, the ordering of the SEMs of each day is quite similar, leading to strong temporal correlation. However, we can notice that sequences involving the same SEM are allowed (whereas they where prohibited for the frequent itemset mining, due to the definition of sets).

Once frequent patterns are found, they are filtered by removing the 1-patterns and the open

sequence size	{sequences}	support
1	$\{home\}$	4
	$\{work\}$	3
	$\{restaurant\}$	3
2	$\{home \rightarrow work\}$	3
	$\{home \rightarrow restaurant\}$	3
	$\{work \rightarrow work\}$	3
3	$\{home \rightarrow work \rightarrow work\}$	3

Table 4.3 – Frequent sequences

patterns. leading to the ordered list of pattern: $\{home, work\}$, $\{home, restaurant\}$, $\{home \rightarrow restaurant\}$ and $\{home \rightarrow work \rightarrow work\}$. Each day is then matched again these pattern resulting in the boolean features described in table 4.4.

Day	boolean features
1	1, 1, 1, 1
2	1, 1, 1, 1
3	1, 0, 0, 1
4	0, 1, 1, 0

Table 4.4 – Boolean features ordered as: $\{home, work\}$, $\{home, restaurant\}$, $\{home \rightarrow restaurant\}$, $\{home \rightarrow work \rightarrow work\}$

Applying a hierarchical clustering of the boolean features, we obtain the following clusters representing CEMs:
$\{day1, day2\}, \{day3\}, \{day4\}$.

4.4 Evaluation

Both finite state machine and frequent itemsets models were evaluated against a real world dataset. The Nokia data challenge dataset [69][72] contains readings from various phone sensors for 191 users. For our study, due to the sparsity of some sensors data, we focused on the GPS traces of the users extended by the GPS location of the wifi routers in range. In order to better focus on the generation of CEMs, we take as input for our algorithms the SEM generated using the modified ESOINN algorithm presented in the previous chapter. The GPS traces are clustered using the modified ESOINN and each cluster is augmented with the semantic of the closest venue returned by Foursquare [49] which was at the time of the experiment one of the largest dataset for semantic representation of locations. At the time of writing, other alternatives such as Google places [56] became more popular and should be preferred. In order to alleviate the absence of feedback from the user, we attribute the label to each cluster once the entire dataset has been processed. This way, we avoid issues such as splitting and merging of clusters that would have required to be handled by the user. Since CEMs represent SEMs with a higher level of abstraction, the evaluation of the CEMs generated using both models was not possible using solely the Nokia dataset. Indeed, the dataset does not contain a ground truth regarding the routines of the users. To solve this issue, we make use of the wisdom of the crowd and we run a crowdsourced experiment, trusting the ability of the crowd to evaluate the CEMs. In addition, to better evaluate our approaches, we compare their results to the ones obtained with the state-of-the-art T-pattern mining algorithm.

4.4.1 Preprocessing

The Nokia dataset contains data about 191 users. Along with GPS information, the dataset defines also visits representing a time window during which the user was considered as being at a single location. However, the users were often using the phone provided for the study as a secondary phone, which was not activated all the time. Furthermore, due to low accuracy GPS reading, some signals are not usable. For this reason, we filter out users with less than at least four days with more that three SEMs, reducing the number of users to 131.

We extend the dataset by adding the home location of each user during each night if this night does not contain any location information. The home location is taken as the most frequent location of the user between 10pm and 7am. The resulting days are then provided to the pattern matching algorithms. The statistics about these days are detailed in table 4.5.

Number of users:	preprocessing: 191 postprocessing: 131
Number of days per users:	$\mu = 33.48$, $\sigma = 33.32$
Number of EEs per day:	$\mu = 4.46$, $\sigma = 0.27$

Table 4.5 – Statistics of the dataset. Days per users and SEMs per day are defined by their average (μ) and standard deviation (σ)

Furthermore, each location record is extended with information gathered from Foursquare (i.e., location name, category, etc.) We based our evaluation on the Nokia dataset (which just contains *When?* and *Where?* information), although MemorySense could leverage further semantic information about the SEM, e.g., the *What?* information. The rationale behind this choice was to test our algorithms on a large, realistic dataset that spanned over many months, at the cost of not leveraging the full potential of the algorithms. We are not aware of any large dataset (i.e., hundreds of users, hundreds of days) which contains *What?*, *When?*, and *Where?*

4.4.2 Generation of CEM

The preprocessing phase has generated a list of days defined as sequences of SEMs for each user. This list is passed on to the frequent pattern mining model and the finite state machine model. At this point, a definition of equality was needed to compare SEMs. A possible formulation of equality between SEMs could be based on the *type* of their venue: an SEM happening in a restaurant in the city center is equal to an SEM happening at the restaurant of the university. In this work we assume a stricter formulation of equality, which means that two SEMs are equal if and only if their *actual venues* are equal. This is motivated by the fact that the automatic retrieval of the venues associated with the stay points computed from the Nokia dataset was not manually validated and therefore our approach could have picked up erroneous patterns due to mislabeled venues. Strict equality prevents these patterns from being detected.

Finite state machine

As defined in the previous section, the finite state machine approach requires a fix number of buckets. We set this number to 24, meaning that each day contains 24 SEMs. Since, the average number of SEMs per day is 4.46, we need to interpolate the representation of the days. This interpolation is done in the following fashion: due to the addition of home locations in the preprocessing phase, we are sure that at least the first bucket of the day (i.e. from midnight to 1a.m.) contains a SEM. From there, if the original day contains an EE happening within a bucket's time slot, this bucket will take the information of the SEM, otherwise, the information of the previous bucket will be used. In case multiple SEMs happen during the same bucket's time slot a majority rule is used to pick the most frequent SEMs.

Once the days are interpolated, they are provided to the finite state machine model that will decide, based on the threshold θ, if the day contributes to an existing FSM or if there is need for a new FSM. We empirically set this threshold $\theta = 5$. The resulting FSMs are taken as the different type of CEMs.

Frequent pattern mining

To evaluate the FPM approach, the list of days is passed to the modified version of the FUP algorithm described above by weekly increments. In other words, groups of 7 days are passed to FUP until all of the days of the user are processed. The minimum support for both frequent itemsets and sequences is set to 7 (i.e. a pattern needs to appear in 7 different days before being labeled as frequent). At the end of each iteration, boolean features are generated for each of the 7 days and are placed in a set containing all the previous features lists. A hierarchical clustering with a maximum distance between clusters of 3 is then performed on this set to group days into CEMs. Although in a real application the cluster to which a day belongs can be determined on the fly when the day is accessed, for the sake of the user study the list of days is scanned one last time to attribute a CEM label to each day. The statistics of the resulting CEMs are provided in table 4.6.

We can note that a day with boolean features $< 0, 0, \ldots, 0 >$ means that it matches none of the frequent patterns. Days having such vector are therefore exceptional and must be treated

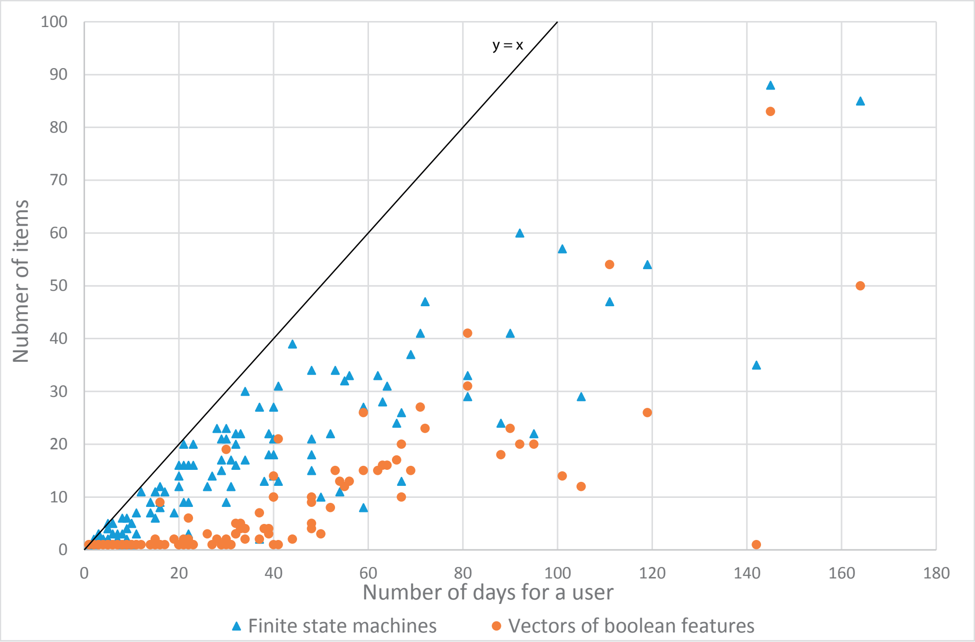


Figure 4.3 – Number of finite state machine created and number of unique boolean features combinations as a function of the number of days per user.

separately. Indeed, these days are very different from one another and should not be used in the clustering phase, instead they are grouped in a separate cluster.

Length of the boolean features:	9.55
Number of unique boolean features:	5.52
Number of CEMs:	3.37

Table 4.6 – Statistics (averaged per user) for the frequent pattern matching approach

The number of finite state machines generated by the FSM model and the number of unique vectors of boolean features created by the frequent pattern mining model for each user are depicted in Figure 4.3 as a function of the number of days collected for each users. We can notice a long tail of users having only 1 vector of boolean features ($< 0, 0, ..., 0 >$). This implies that no frequent patterns were discovered for these users. This behavior is an artifact of the nokia dataset and happens essentially with users with not enough data and irregular routines: 97% of these special cases have less than 30 days of data.

4.4.3 Comparison with T-patterns

In order to further evaluate the two approaches, we compare them to the T-pattern mining algorithm developed in [53]. The algorithm takes as input raw GPS traces and produces temporal pattern of the form $location_1 \xrightarrow{\Delta_1} location_2 \xrightarrow{\Delta_2} ...$ where $location_i$ represents what Giannotti introduces as region of interest and can be assimilated to the Place of Interest defined in this thesis. The Δ_i provide temporal information regarding the waiting time before observing a transition between $location_i$ and $location_{i+1}$.

We selected this algorithm for 2 reasons: 1/ This algorithm is based on the density of the GPS signal to define regions of interest, which follows a similar approach to our generation of PoI. it is therefore more convenient to compare both approach. 2/ T-pattern proposes to detect unbounded patterns and therefore offers the same modelling flexibility than our approaches.

The T-pattern algorithm is base on level-wise approach similar to the GSP algorithm, in the sense that it builds frequent patterns of increasing size base on a minimum support threshold. The parameters of the algorithm as defined in Table 4.7. The lines of the algorithm are taken from [53] and provided in Algorithm 5. It uses the following functions: $ExtractFrequentTimings(T)$ which generate temporal annotations for trajectories T, $pruneEmptyAnnotations(T, A)$ which removes trajectories that did not have any temporal annotations, $Density(T, \mathscr{G}_0), \epsilon)$ which generates a grid with cell densities reflecting the number of trajectories visiting it, $PopularRegions(\mathscr{G}, \delta)$ which extract regions of interest, $translate(T, RoI)$ which transform trajectories into sequences of RoI and $generateLargerTrajectories(D, A, prefix, r)$ which builds prefixes of size $L+1$.

Parameters	**Description**
$\mathscr{G}_0$	A grid over GPS space to discretise it
δ	Minimum support threshold
ϵ	radius for spacial neighbourhood
τ	temporal threshold

Table 4.7 – T-pattern parameters

We use the open-source implementation of the T-pattern algorithm, and we choose its parameters to match the conditions of both our approaches: an absolute minimum support and density threshold $\delta = 7$, a grid subdivision $\mathscr{G}_0$ of $\frac{1}{100}$ of the boundary on the GPS traces, a time limit τ infinite, and a default ϵ radius for spacial neighbourhood (0). There is therefore no time constraint for the generation of T-patterns.

The resulting T-patterns that the algorithm provides are then used to cluster the days of the user into routines. The days are then semantically enhanced based on the regions of interest define in the T-pattern in order to have a similar representation than the output of our algorithms. This ensures that the following crowdsource evaluation could handle indistinguishably the three system FPM, FSM and T-pattern.

Algorithm 5 T-pattern algorithm

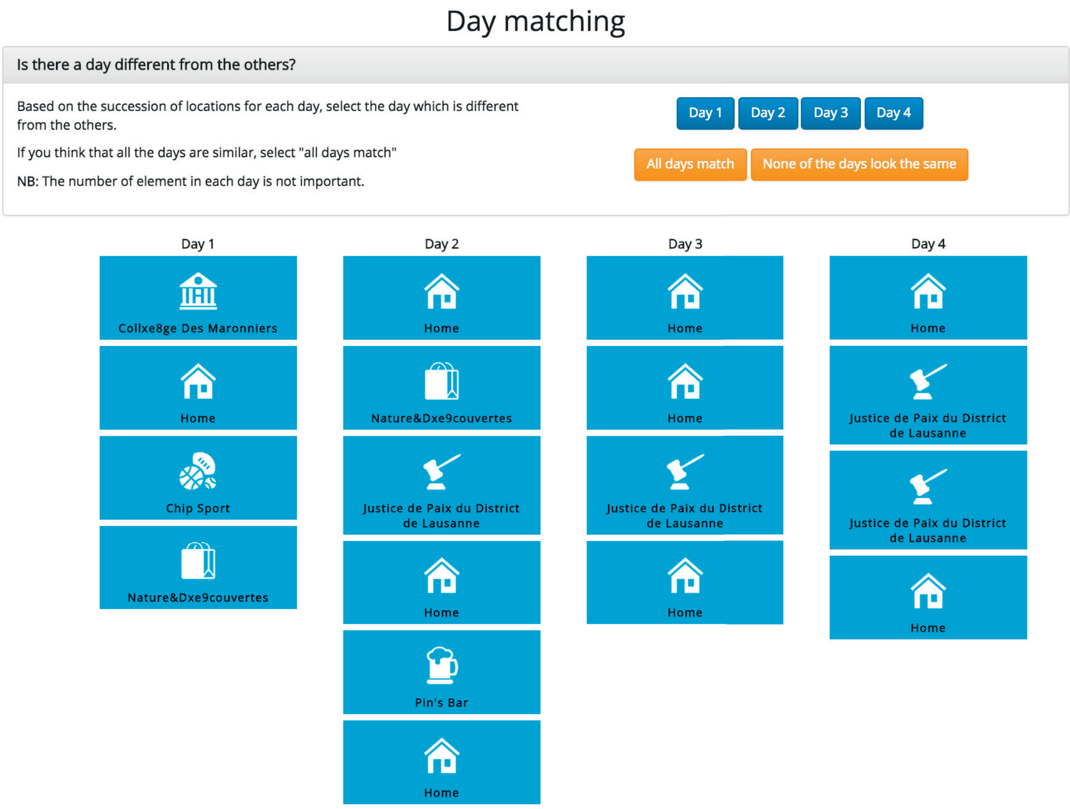
```
Input: $T_i n, \mathcal{G}_0, \delta, \epsilon, \tau$
Output: $O = \{(S, A)\}$ //set of pairs of sequences of regions and their associated temporal transitions
$L = 0$
$T_0 = \{(T_i n \times \{\emptyset\}, <>)\}$
while $T_L \neq \emptyset$ do
    $T_{L+1} = \emptyset$
    for $(T, prefix) \in T_L$ do
        if $len(prefix) > 1$ then
            $A = ExtractFrequentTimings(T)$
            $O = O \cup (prefix, A)$
            $T = pruneEmptyAnnotations(T, A)$
        $\mathcal{G} = Density(T, \mathcal{G}_0), \epsilon)$
        $RoI = PopularRegions(\mathcal{G}, \delta)$
        $D = translate(T, RoI)$
        for $r \in RoI$ do
            if $support_D(r) > \delta$ then
                $T_{L+1} = T_{L+1} \cup generateLargerTrajectories(D, A, prefix, r)$
    L++
```

4.4.4 Crowdsourced evaluation

Since the Nokia dataset does not provide a ground truth regarding the routine of the user, we run a user study on Amazon Mechanical Turk [8] to assess if the routines identified by T-patterns, FPM and FSM are deemed reasonable by humans. This approach reminds of the one presented by Chang et al. [22], in which the authors leverage the crowd to analysis the pertinence of topic models which are acquired in an unsupervised fashion. The authors propose to the crowd several tasks in which the workers are asked to identify outliers among sets of words related to a topic, or sets of topics related to a document.

In a similar fashion, we design an experiment in which each worker is presented with 4 different sets of daily activities, and 6 different choices. The choices are the following: (*Outlier*) 4 buttons for each single day, to signal that the given set of activities was an outlier among the other 3; (*All-match*) 1 button to signal that all the 4 days could be mapped to the same routine; (*No-match*) 1 button to signal that all the 4 days were different (therefore not mappable to the same routine). All the tasks have been generated automatically, in a uniformly randomised fashion among 3 scenarios for each approach:

- The 4 days presented to the worker belong to the same routine
- 3 of the 4 days presented to the worker belong to the same routine and a last day is selected from another routine (the place of the outlier is randomised)
- The 4 days belong to 4 different routines



Day matching

Is there a day different from the others?

Based on the succession of locations for each day, select the day which is different from the others.

If you think that all the days are similar, select "all days match"

NB: The number of element in each day is not important.

Day 1
Day 2
Day 3
Day 4
All days match
None of the days look the same
Day 1
Collxe8ge Des Maronniers
Home
Chip Sport
Nature&Dxe9couvertes
Day 2
Home
Nature&Dxe9couvertes
Justice de Paix du District de Lausanne
Home
Pin's Bar
Home
Day 3
Home
Home
Justice de Paix du District de Lausanne
Home
Day 4
Home
Justice de Paix du District de Lausanne
Justice de Paix du District de Lausanne
Home

Figure 4.4 – Crowdsourced experiment interface

This task is simple as compared to other tasks requiring specific knowledge or training from the worker. For this reason, we did not impose any restriction on the workers to which the tasks were proposed. The web interface presented to the users is displayed in Figure 4.4. It displays a small description of the task to guide the workers, the 6 action buttons and the 4 days to evaluate. The representation of the days is a list of PoIs visited by the user. Each PoI is displayed using the semantic data provided by the Foursquare for this place. In order to alleviate the fact that some places had non english labels (due to the fact that the data has been acquired essentially in Switzerland), we also provide the workers with the icon associated to the PoI in order to have a visually intuitive way of finding routines.

In total, combining the 3 different approaches, more than 3800 tasks, paid \$0.02 each, were performed by 279 crowd workers. Contrary to standard classification tasks in which the crowd is used to determine a ground truth, the goal of our crowdsourced evaluation was rather to assess if the workers were finding the routines meaningful. For this reason we did not aggregate results of repeated comparisons. Due to the comparatively small size of our study, we did not receive any feedback from the users on the forum associated with the task.

Table 4.8 reports the results in term of accuracy sensitivity and specificity that we obtained from the crowd, with FPM as a clear winner. Given the 6 choices presented to the user, on the same task a randomised baseline algorithm would perform with a 16.7% accuracy. Realistically though, an algorithm that performs poorly could obtain way worse results than the randomised baseline in an evaluation with human subjects, e.g., the algorithm could fail to identify many important routines (therefore classifying most of the days as *No-match*), while they will not go unnoticed to the crowd workers.

It is also interesting to notice that a state-of-the-art algorithm like T-patterns performs almost as the randomised baseline for this specific task, confirming that MemorySense is tackling a hard variant of a well-known class of problems. Such performance does not question the validity of T-patterns per se, but only its suitability in the context of identifying user routines.

	Accuracy	Sensitivity	Specificity
T-patterns	21.6%	40.6%	8.8%
FSM	44.0%	59.1%	49.7%
FPM	56.3%	74.3%	51.9%

Table 4.8 – Crowdsourced evaluation of T-patterns, FSM and FPM

The accuracy has been measured according to the confusion matrix in Table 4.9.

	Outlier	All-match	No-match
Outlier	**32.1** / (20.3)	12.7	3.8
All-match	10.3	**7.0**	1.1
No-match	0.8	0.7	**11.2**

Table 4.9 – Confusion matrix for the classification tasks performed by the crowd (compound of T-patterns, FSM and FPM). All numbers are percentages, and the correct classifications are reported in bold. Rows are expected answers, columns are the input from the crowd.

For the sake of readability, we compacted the *Outlier* 4−categories classification task as a single entry in the matrix. The first cell contains in bold the percentage of outliers that have been correctly identified among all the tasks submitted to the workers, while the value in parenthesis represents the mislabeled outliers. All the other cells follow the usual behaviour of a confusion matrix, with the diagonal reporting the accurate classifications, and all the other cells reporting the misclassifications.

4.4.5 Performance evaluation

In order to assess that our approach is suitable for nowadays devices, we also evaluated the runtime performance of both SEM generation methods. We used a Google Nexus 4 with a quadcore CPU (1.5 GHz), 2GB of RAM and the operating system Android 4.4.

Both finite state machine and frequent pattern mining approaches are compared to a baseline defined as a non-incremental version of the frequent pattern mining process, in which fre-

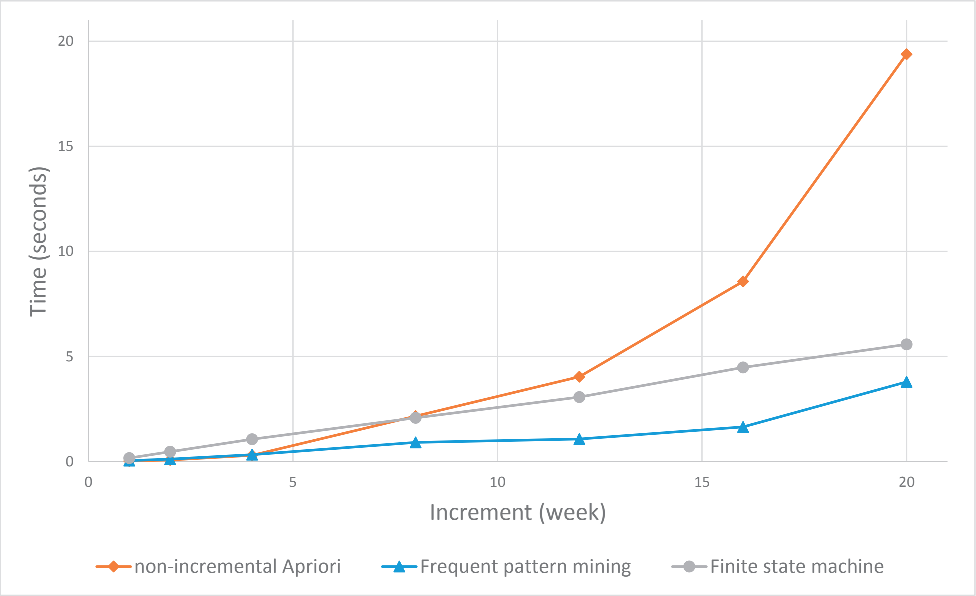


Figure 4.5 – Time required to process a new week of data for both SEM generation methods, compared to a non-incremental method based on original Apriori and GSP.

quent itemsets are discovered using the original Apriori algorithm, and frequent sequences are discovered using the original GSP algorithm. Their relative minimum support is adapted for iteration to simulate an absolute minimum support of 7. The rest of the process (i.e., boolean features generation and hierarchical clustering) remains as described in the previous section.

The results presented in figure 4.5 show that, as weeks accumulate, the processing time required by the non-incremental method outgrows significantly our two approaches, justifying the incremental processing design choice of MemorySense. The time taken to process a single week of data is below 10 seconds for both our algorithms, therefore an app that embeds MemorySense will not make the phone unresponsive, nor it will hinder the user experience.

Given that the battery consumption is influenced by the CPU usage, it is also worth noticing that the incremental processing capabilities of MemorySense make it a battery-conscious framework – an especially desirable feature, considering the plethora of energy-hungry applications that run on modern smartphones.

4.5 Conclusion

The system presented in this chapter aims to model, in an energy efficient manner, the past activities of a user, similarly to the way the human brain builds memories. We propose two concurrent methods to aggregate low level pieces of memories (SEM) into more abstract complex episodic memories. these models have the capabilities to handle variation of routines such as swapping the order of two SEMs, deleting an SEM, replacing it or adding an extra one. This flexibility can be seen as a filter that remove the semantic noise present in users' routines. The first approach, based on finite state machine, presents a discretisation of the day and builds models of CEM as a direct graph containing all the variations of a same routine. The second approach, based on frequent patterns focuses on detecting frequent set of SEM in routines (for which the ordering is irrelevant), as well as frequent sequences which focus on reenforcing time dependant patterns in CEM.

To compensate for the lack of information about CEM in the dataset used for the experiment, we conducted a crowd-sourced experiment to evaluate the soundness of our approach. The results show that we improve almost 3 times on the results obtained with the T-pattern algorithm, without leveraging any user feedback. The performance evaluation also shows that both our methods for generating CEMs are suitable for mobile computation.

Glycemic State Recognition Part II

5 Background

5.1 Introduction

Diabetes type 1 patients are not able to produce insulin. This hormone, which is synthesised by the β-cells of the pancreas, is essential for the regulation of the glucose level in the blood. The lack of insulin induces a rise of the glucose level, which can lead to complications. In order to regulate their glucose level, diabetic patients have to inject herself with insulin regularly. The amount of insulin to inject depends on various factors such as their current glucose level, their food intake, the amount of insulin already injected, and their physical activities. Overestimating this amount will result in state of low blood glucose level (Hypoglycaemia), while underestimating it will result in high glucose level (hyperglycaemia). Injecting the correct quantity of insulin will keep the glucose level within acceptable range (euglycemia). However, this estimation is difficult to achieve and as a result, many patients suffer from hyper- and hypoglycaemic episodes on a regular basis. Consequences of Hypoglycaemia can be dramatic, ranging from headaches to judgment impairment and loss of consciousness, while consequences of hyperglycaemia are more related to long term effects such as nerves or kidney damages.

The last century has seen an increase in type 1 diabetes [51], as well as type 2. This raise triggered a wave of research aiming to improve the quality of life of diabetics. Measuring the blood glucose level can currently be done in two different ways: (i) Using a glucometer which directly measures the blood glucose level; (ii) Using a continuous glucose monitoring system (CGMS) which measures the level of glucose in the interstitial tissues and deduce the glucose level from it.

In the recent years, the advent of wearable devices led to the notion of quantified self, which relates to the evaluation of one's physical parameters such as activity level, heart rate, weight, or temperature. However, the sensors data collected by these wearable devices can be used to infer much more information about the user.

Indeed, it has been shown by Laitinen et al. [71] that the shape of the ECG can be used to determine if a patient is in an hypoglycaemic state or not. Furthermore, physical activity can lead to hypoglycaemia if the patient does not control her blood glucose properly [13]. This measure can be extracted from wearable devices sensors data, and can therefore be used to evaluate blood glucose level.

In this part of the thesis, we present two methods to infer the user's glucose level based solely on sensors readings: a physiological models based on the electrocardiogram (ECG) of the patient, and a model, based on her energy expenditure. Both models are using multiple sensors data coming from the off-the-shelf sport belt: Zephyr Bioharness [128] (Figure 5.1). This belt allows to follow the user through the day and to collect 3 type of signals: ECG signal representing the heart electrical activity (sampling rate), breathing signal measuring the extension of the rib cage, and 3 axis accelerometer.

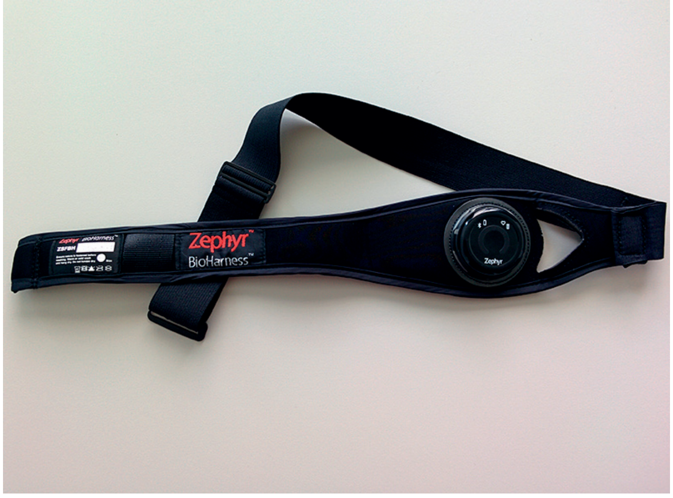


Figure 5.1 – Zephyr bioharness sensor belt

The contribution of this part is the development of two methods to estimate the glucose level of diabetes type 1 patients based on wearable sensors data, in an everyday life settings. An off-the-shelf sport belt was used to acquire the data. This setting implies that our models handles realistic, but highly noisy, data. The evaluation however, shows that our method perform on par with the state of the art that uses data acquired in an hospital environment.

5.2 General information

5.2.1 Diabetes

Diabetes (Diabetes mellitus) is a group of chronic diseases that translate into the inability for the body to properly regulate the glucose level in the blood. We can differentiate 3 types of diabetes:

- Gestational diabetes may arise during pregnancy when the mother-to-be displays high blood sugar.

- Diabetes type 1 refers to the case in which the β-cells of the pancreas - which are responsible for producing insulin - do not fully fulfil their function, leading to a lack of insulin in the blood and a rise of the glucose level.

- Diabetes type 2 refers to the case in which the cells present a resistance to insulin, therefore attenuating its effect. This form of diabetes can later lead to a failure of the β-cells to produce insulin.

The populations affected by these diabetes are diverse. The gestational diabetes, as it names implies, essentially occurs during pregnancy and usually goes down once the baby is born. In the case of diabetes type 1, several genes have been identified as influencing the risk of diabetes, but little is known about the decisive factors in the development of diabetes type 1. However, the development of this diabetes usually take place during childhood and can affect children indifferently from their diet, body mass or height. In comparison, Diabetes type 2 appears around the adult age as it is mainly attributed to lifestyle factors such as lack of exercise and unhealthy diet.

Glucose fluctuation is natural in human metabolism. The glucose level is globally low after fasting and rises after eating due to the digestion. The main source of glucose are the carbohydrates (molecules composed on Hydrogen, Carbon and Oxygen), such as sucrose (the standard sugar that can be used to sweeten food), which are metabolised by the organism. In other words, glucose is the result or breaking down and reassembling larger molecules found in food. Additionally, glucose can also be created by gluconeogenesis. This mechanism represents a generic metabolisation that results in the production of glucose. It can result from the breakdown of proteins as well as lipid and account for a small percentage of the glucose level.

Additional factors such as physical activity impact the glucose level. However, the amount of insulin produced by a non-diabetic person is automatically regulated in such a way that will keep the glucose level within a range defined as normal (euglycemia).

For the 3 types of diabetes, if no counter-measure is applied, the lack of insulin, or the resistance of the body to it, will inhibit this control mechanism, leading to a range of high glucose level (hyperglycaemia). In order to keep the glucose level within acceptable range, diabetics have several levers than need to be adequately activated. While for gestational and type 2 diabetes regular physical activity and a strict diet can mitigate the resistance to insulin, diabetics type 1 simply does not produce enough of it. Therefore, on top of the physical activity and the diet, diabetics type 1 also need an external source of insulin in the form of shots directly injected in the bloodstream. However, overshooting the amount of insulin injected will result in an episode of hypoglycaemia.

5.2.2 Insulins

Diabetics type 1 inject with insulin to regulate their glucose level. There exist different types of insulins with different activation times and therefore different purposes:

Fast acting insulin. Fast insulin (e.g. NovoRapid from the company Novonordisk) is effective 10 minutes after the injection and lasts for a duration of 2 hours. This type of insulin is used to cut the excess of glucose. It is usually injected before meal in order to counter-balance the glucose ingested, but can also be injected upon detection of high blood glucose level.

Long acting insulin. The long acting insulin, also known as basal insulin (e.g. Tresiba from Novonordisk) has a duration of action of more than a day (more than 42 hours in the specific case of Tresiba). Once injected, this insulin aggregates to form reserves of multiple molecules of insulin, which are then slowly released in the body. This insulin is used by some diabetic patients to serve as a nominal amount of insulin. This insulin limits the level of glucose of a standard day. Fast acting insulin can be used on top of the basal insulin in order to flatten further the glucose variations. The long lasting insulin is usually taken every couple of days and regardless of the activity or diet of the patient.

Short and intermediate acting insulin. Short and intermediate acting insulin complete the spectrum of insulin variants by having onset times of respectively 30 min and 1 hour and durations ranging from a couple of hours for short acting insulin to 18 hours for intermediate acting insulin. While short acting insulin is mainly used in anticipation of meals, intermediate acting insulin be prescribed to help half day or overnight glucose regulation.

The release of the insulin in the body is non linear. Fast acting insulin exhibits a peak approximately an hour after injection. At this peak, the amount of insulin that can counteract the glucose present in the blood is maximal. The level of long acting insulin remains comparatively flat throughout the day.

An alternative to manual insulin injection is to use an insulin pump (c.f. Figure 5.2). This device is composed of a catheter that is placed under the skin and linked to the pump device. The pump is programmed to release continuously small quantities of short acting insulin, playing the role of a basal insulin. When properly calibrated, the pump minimise the fast variations of insulin in the blood. The pump can also release more insulin upon request from the patient, for instance for the meals or when blood glucose level is high.

5.2.3 Glucose monitoring

In order to evaluate their need for insulin, diabetic patients need to perform regular glucose measurements. This is currently done through two invasive methods: Using a glucometer or a continuous glucose monitoring system (CGMS).

A glucometer is device used to manually measure one's blood glucose level. As depicted in

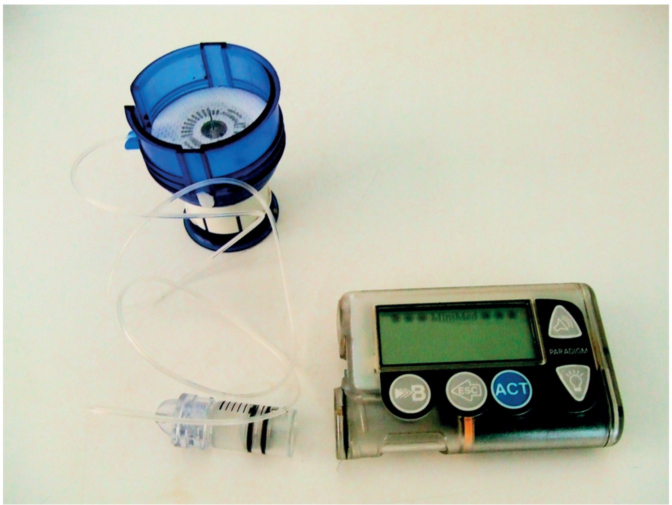


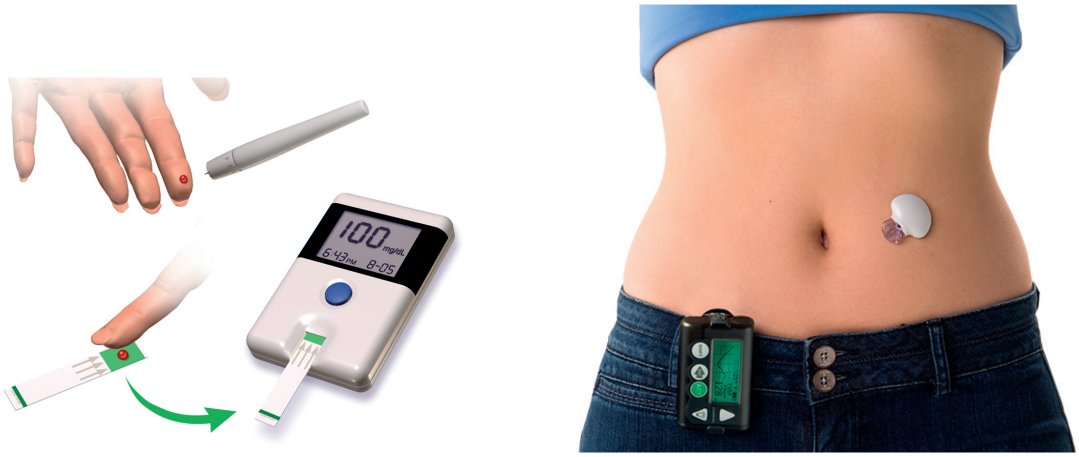
Figure 5.2 – Insulin pump (Source: wikimedia)

Figure 5.3a, a lancet is used to puncture a finger and draw a drop of blood. The blood is placed on a strip which is then inserted into the glucometer device. The device then performs analysis of the blood and displays the concentration of glucose detected. This type of measurement is considered a gold standard as it measures the quantity of glucose directly in the blood. However, this process is cumbersome and invasive as it necessitates a new puncture for each measurement. For this reason, it is only used 5 to 10 times per day, usually before meal to know the amount of insulin to inject and some time after meal to control the glucose level.

In order to perform a more continuous evaluation of the glucose level, diabetics can use a CGMS. As depicted in Figure 5.3b, a sensor is placed directly on the skin and a catheter is implanted under the skin to capture the concentration of glucose present in interstitial fluids. This value is used as a proxy for actual blood glucose concentration. The sensor value is then stored on device or wirelessly transmitted to a monitor. CGMSs are used to have a more fine grain representation of the patient's glucose level, and is used for instance in conjunction with insulin pump to evaluate the adequacy of the insulin program or to study glucose levels overnight. However, CGMS need to be calibrated using a standard glucometer several times per day, making it also a cumbersome solution.

Once the patient knows her glucose level, she needs to estimate the correct amount of insulin to inject. This is not an easy task and it is common for diabetic patients to miscalculate the dose of insulin that needs to be injected, leading to hypoglycaemia, and hyperglycaemia. In the following chapters, we use the following definitions of such events:

- Hypoglycaemia is considered to occur when the blood glucose level drops below 3.9 mmol.l^{-1}.
- Hyperglycaemia is defined for a blood glucose level above 7.8 mmol.l^{-1}.



(a) Glucometer (Credits: Blausen.com staff. "Blausen gallery 2014". Wikiversity Journal of Medicine.)

(b) Continuous glucose monitoring system Ipro2 (Credits: Medtronic)

Figure 5.3 – State-of-the-art devices for blood glucose level monitoring

- Euglycemia is defined as a normal blood glucose level between hypoglycaemia and hyperglycaemia.

5.3 State of the art

The work on improving glucose management system has recently soared. In 2014, Google and Novartis joined to develop a glucose sensor base on smart contact lenses [108]. The year 2016 witnessed the beginning of a partnership between IBM Watson and Medtronic (one of the leader in diabetes management), in order to harness IBM Watson's analysis abilities to advance research on diabetes treatment. Medtronic received the approval of the FDA (Food and Drug Administration, the US regulation organ for new medical devices and drugs) for its MiniMed 670G ([109]), the first closed-loop hybrid insulin pump. Used in conjunction with a CGMS, the device automatically adjusts the amount of insulin to inject to the user.

We analyse the most relevant works related to the issues addressed in this part:

- Non-invasive methods for monitoring glucose levels
- Energy expenditure estimation techniques.

5.3.1 Non-invasive glucose monitoring

Recent years have seen the development of techniques aiming at replacing invasive methods such as the use of glucometer to evaluate the glucose level. In [39], Ding et al. present a comprehensive review of the use of mobile sensors to infer physical activities, with the goal of improving glucose management for diabetic patients. Another review of the different means that can be used (not necessarily via physical activity detection) to evaluate glucose level is also presented in [103].

In [31, 30, 19], the authors leverage signals such as accelerometer data and heart rate in order to refine the insulin dosage provided by pumps and artificial pancreas. For the same purpose, in [118], the author uses energy expenditure and galvanic skin response to estimate the food intake along with glucose measurements. In [126, 127], Zanon et al. describe a multisensory approach to glucose level evaluation based on the dielectric property of the skin. The author leverages dielectric spectroscopy to measure these changes and uses the readings from additional sensors such as skin temperature, humidity, accelerometer data, to alleviate the noise from the dielectric measurements. Additional methods of indirect tracking of the blood glucose level can be found using sweat ([110]), saliva ([73]) or urine for intermittent glucose monitoring, or the tracking of body tissues using either transdermal or optical characteristics of the tissues to infer blood glucose level ([67]). Plis et al. [93] propose a machine learning approach to 30 minutes and 60 minutes prediction of the glucose level based on glucose level history and manual annotations from the user regarding her insulin intake, meal and activity. The authors use a physiological model to build informative features about the user and then use support vector regression to estimate the evolution of glucose.

Closer to our concerns, Stenerson et al. in [115, 114] leverage a Zephyr bioharness sport belt ([128]) to augment a low glucose prediction algorithm with accelerometers and heart rate data. They investigate the use of a Kalman filter with 30min prediction window to further reduce hypoglycaemic events. In [111], soltanzadeh proposes a customised portable ECG and envisions its use for diabetes management.

Approaches using machine learning to detect hypoglycaemia have also been studied extensively by Hung Nguyen and his team. The author published a important amount of papers in the last two decades regarding the use of ECG and, in a lesser measure, EEG (Electroencephalogram) to detect glycemic events. Nguyen uses neural network-based systems to classify the glycemic events using physiological features of the signals (selection of papers presenting these techniques [21, 76, 104]). In [75], the author trains a support vector machine (SVM) model to detect hypoglycaemic event from ECG signal using similar features to our approach. Phyo P. San presents several approaches [105, 77] to detect hypoglycaemic events using neural networks. For instance, in [105], the author trains a deep neural network on ECG signals to detect Hypoglycaemia. His results are considered the state of the art, and are taken as reference to evaluate our physiological approach.

Finally, models simulating the glucose-insulin control system have been proposed in order to develop closed-loop glycemic control, for artificial pancreas. Herrero et al. [63] built a controller to automatically regulate the insulin pump of an artificial pancreas. This controller uses a mathematical model inspired from the functioning of β-cells of the pancreas. The approach shows good results for alleviating glycemic events. However, the authors emphasises that the delays resulting from sensing glucose and releasing insulin are the main challenges of closed loop glucose regulation systems. Dalla and Cobelli proposed in [36] a simulation model to study postprandial events related to glucose and insulin levels. Since then, several approaches have been derived from this work. For example [74] proposes to simplify Cobelli's model to allow its computation on FPGA devices. In [35], Dalla and Cobelli augment the simulation model by taking into account physical activities, and using the heart rate as a proxy to evaluate this activity. The evaluation is performed using *in silico* trials (i.e. personalised simulation).

The idea behind our approach is to compare the glucose intake through food (we are not mentioning the extreme case of direct glucagon injection, but a similar reasoning could apply), with the glucose usage through energy expenditure and glucose neutralisation through insulin injection.

5.3.2 Energy expenditure

The energy expenditure is primordial when building our model based on energy balance. In the remainder of this subsection, we review the literature related to the task of estimating the physical activity and energy expenditure of users based on wearable sensors. An important amount of work has been done trying to characterise physical activities from sensor data. A great majority of the approaches use accelerometers, which are nowadays embedded in all smartphones [60, 70, 62]. In [70], for instance, the authors present a method for inferring the activity of the owner of a smartphone using its accelerometer data. The study focuses on 6 basic activities (running, walking, standing, sitting, going up and down). Another approach [123] proposes a similar feature but with a more energy efficient approach. Alternatively, dedicated devices have also been used to embed accelerometers to detect activities at a coarse level, using wrist and arm bands, for example [81, 41]. Other type of sensors such a camera have also been used. In [86], the author uses the optical flow (motion of object/character in a video) of the person, in order to infer her activity. While these approaches allow an efficient characterisation of the user's position, they lack the granularity required to define the intensity of the activity. For example, running can be done at different speed and under various conditions.

Closer to our concerns, there exist also several approaches in the literature to derive the energy expenditure (EE) via indirect calorimetry. While these approaches do not discriminate among activities, they provide a good indicator of the amount of energy that is being used. Doubly labeled water [107], is considered the gold standard for measuring EE. It makes use of isotopes of oxygen and hydrogen in order to mark water molecules ingested by the organism. The carbon dioxide released by the organism comes from the metabolism (i.e. it is a by-product of the energy produced by the organism). Therefore, it is possible to measure the energy expenditure. Alternatively, various approaches [98, 29] use the amount of oxygen breathed by the subject as a proxy for the intensity of the physical activity. These quantities vary also with the intensity of the activity and are considered reliable to estimate the energy produced.

However, doubly labeled water and VO2 (oxygen consumption) are not convenient for continuous monitoring as they necessitate either regular sampling of body fluid, or special equipment that is not wearable. Altini et al. [7] studied the role of the positioning and number of accelerometer sensors in the estimation of EE. The authors report that a single sensor placed close to the waist or chest is to be preferred to estimate EE using activity specific models. In [88], the author uses accelerometer data to infer the MET (Metabolic equivalent of task), while in [18] and [106], heart rate is combined to accelerometer to define a level of activity. Cvetkovic et al. [33] evaluates separately accelerometers and heart rate + breathing rate + near body temperature for EE estimation and concludes that. In [11], the authors use an armband fusing galvanic skin response, skin temperature, near-body temperature, heat flux sensor and accelerometers to define the EE in children. However, using a combination of breathing rate, heart rate and accelerometer in an everyday life setting has not been studied so far.

We propose in the rest of this part two approaches in order to detect these events. The first one, based on physiological symptoms, aims to detect hypoglycaemia, while the second approach aims to build a model of the blood glucose level of a user and be able to evaluate at any point in time her blood sugar level.

6 Physiological model

6.1 Introduction

During a period hypoglycaemia, patients can experience various apparent symptoms such as sweat, headache and shakiness, as well other symptoms related to the central nervous system such as mood swings. Other symptoms, less apparent occur. One of them is the alteration of cardiac cycle. In this chapter we present a physiological model that relies essentially on ECG features to detect hypoglycaemia in a mobile environment. Using ECG characteristics to infer hypoglycaemic events has already been studied. Hung T. Nguyen, for example, proposed several artificial neural network approaches along this line ([105, 104]). However, to the best of our knowledge, the data used in these approaches is collected in an hospital environment using medical-grade devices. Although the quality of the acquired signal is very good, it does not reflect the every day life conditions of the patients. In this chapter we propose to evaluate a more pragmatic approach by using an off-the-shelf sensor belt: Zephyr bioharness (Figure 5.1) that the user wears while performing normal life activities. Pursuing the goal of preserving patients privacy, we perform all data processing directly on a mobile device that receives data from the belt. This setting implies that the system needs to handle noisy signals in a limited environment. For this reason, all the algorithms presented in this chapter have been selected for their incremental processing, as well as their general performance.

6.2 ECG alterations

In [71], the author the effects of hyperinsulemic hypoglycaemia on the ECG. This type of hypoglycaemia is caused by an excess of insulin in the blood, leading to a decrease of the blood glucose. It is particularly common among diabetes type 1 patients who do not manage to regulate their glucose correctly. The author explains that hypoglycaemia impacts the various phases of a heart beat cycle, specifically the atrioventricular conduction, ventricular depolarisation and ventricular repolarisation, and these alterations can be observed through their respective ECG consequences. It is therefore possible to identify a list of features that can be

name	description
R amplitude	The ECG value at the R peak
T amplitude	The ECG value at the T peak
ST elevation	The amplitude of the ECG 60ms after the QRS complex (i.e. between S and T)
QT c-Fridericia	The time interval between Q and T of the same beat, normalised by the heart rate (R peak frequency)
T area	The area under the T wave (integral)
R area	The area under the R wave (integral)

Table 6.1 – Features allowing to differentiate ECG under hypoglycaemia and euglycaemia

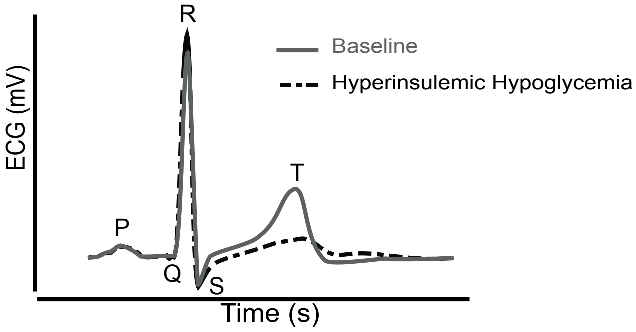


Figure 6.1 – Fiducial points (PQRST) of a heart beat on an ECG representation. Adapted from Laitinen [71].

used to discriminate heart beats of a diabetic patient under hypoglycaemia and euglycaemia. This list can be found in Table 6.1.

These features are described in term of the fiducial points of the ECG. Fiducial points are the key points that characterise a heart beat. They are defined as P, Q, R, S, T labels and can be seen in Figure 6.1.

6.3 Physiological Model

6.3.1 Adaptive filtering

The ECG signal acquired from the belt is of high frequency (250Hz) and suffers from noise related to the movements of the user, either from the friction of the belt on the body, or from other electrical noise from the body itself. While the later is hard to filter, the former can be adequately addressed using an adaptive filter to leverage the correlation between the accelerometer signal and the noise of the ECG to effectively cancel the noise.

Adaptive filter, corresponds to a class of linear filters which transfer function can be optimised, as compared to standard linear filters. As an adaptive filter, the least mean square filter (LMS

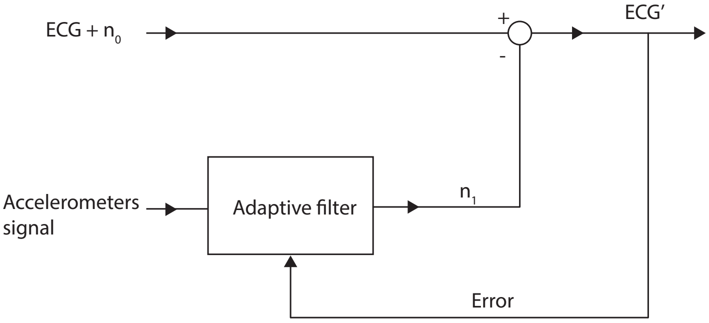


Figure 6.2 – Adaptive filter structure

filter) takes as input two signals:

- A primary input that contains the desired signal and some noise. In our case, the desired signal is a clean ECG and the noise is the interferences coming from motion
- A reference input. This input is correlated to the noise present in the primary input. In our case we consider 3 reference inputs coming from the 3-axis accelerometer

As described in Figure 6.2, in order to perform the filtering, the reference input is processed by a linear filter of order o with weights $\vec{w}$ (an o dimensional vector) to output what the filter think is a good representation of the noise present in the primary input: n_1. This estimation of the noise is then subtracted from the primary input $ECG + n_0$ to produce the filtered signal $ECG' = ECG + n_0 - n_1$. The key idea of the LMS is to reuse the filtered signal as input to optimise the weights w of the linear filter. As its name imply, this optimisation is performed using a stochastic gradient descent based on the least mean square error of the filtered signal and controlled by a learning rate μ.

However, LMS has the drawback of being sensitive to the scaling of the reference input. Since the update of the filter's weights is a function of this input, variations in its scaling will effectively result in variations of the learning increment. This problem is solved by the normalised least mean square filter (NLMS) which normalises the reference input by its power ensuring more stable learning increments. Due to this property, we implemented NLMS for D1namo. However, the signals provided by sensor belt have different sampling rate. The ECG signals has a sampling rate of 250Hz, while the accelerometers have a sampling rate of 100Hz. This would lead to complication when performing the filtering as every signals are expected to have the same frequency. For this reason, we first perform a linear resampling on the accelerometer signal to 250Hz. Given the known accelerometer datapoint (t_0, x_0) and (t_1, x_1), we interpolate the value of the accelerometer at time $t_0 < t < t_1$ by the following equation:

$$x(t) = x_0 + \frac{x_1 - x_0}{t_1 - t_0} * (t - t_0) \tag{6.1}$$

Once the resampling performed, the filter can be applied. The successive steps of the NLMS filtering are presented in Algorithm 6 using the following notations:

- $\overrightarrow{primary}(n) = \overrightarrow{ECG}(n) + \overrightarrow{n_0}(n)$ the noise ECG signal at time n, taken as the primary input.
- $\overrightarrow{acc}(n) = [acc(n), acc(n-1), ..., acc(n-o-1)]^T$ the acceleration signal at time n (each of the 3 axis are handle using different filters). We consider it as the reference input.
- $\overrightarrow{w}(n) = [w(n)_0, w(n)_1, ..., w(n)_{o-1}]^T$ the vectors of weights of size o after n updates.
- $\overrightarrow{n_1}(n)$ the vectors of weights of size o after n updates.
- $\overrightarrow{ECG'}(n)$ the filtered ECG signal at time n.
- $\overrightarrow{X}^H$ represents the hermitian transpose of $\overrightarrow{X}$ (this is equivalent to a regular transpose in the case of real vectors)

We apply the NLMS with the following parameters $\mu = 0.2$ and $o = 50$. An example of the resulting filtered signal is displayed in Figure 6.3.

Algorithm 6 Adaptive filtering with NLMS

```
Input: $o, \mu, \overrightarrow{primary}(n), \overrightarrow{acc}(n)$
Output: $\overrightarrow{ECG'}(n)$ //Filtered ECG signal
  $\overrightarrow{w}(0) = zeros(o)$
  for $n = 0, 1, 2, ...$ do
    $\overrightarrow{n_1}(n) = \overrightarrow{w}(n)^H * \overrightarrow{acc}(n)$ // computes the output of the filter
    $\overrightarrow{ECG'}(n) = \overrightarrow{primary}(n) - \overrightarrow{n_1}(n)$ // computes the filtered ECG
    $\overrightarrow{w}(n+1) = \overrightarrow{w}(n) + \mu * \frac{\overrightarrow{ECG'}(n)^* \overrightarrow{acc}(n)}{\overrightarrow{acc}(n)^H \overrightarrow{acc}(n)}$ // update the filter weights based on a normalised version of the reference signal
```

6.3.2 mathematical morphology

Once the ECG is filtered, we use the approach defined by Yazdani et al. [124] to extract the QRS fiducial points incrementally. This approach is based on mathematical morphology (MM) operators. These operators are usually applied to digital images to perform a succession of operations in order to filter the image for example to enhance the image before performing an optical character recognition. In its core, MM contains two essential operators:

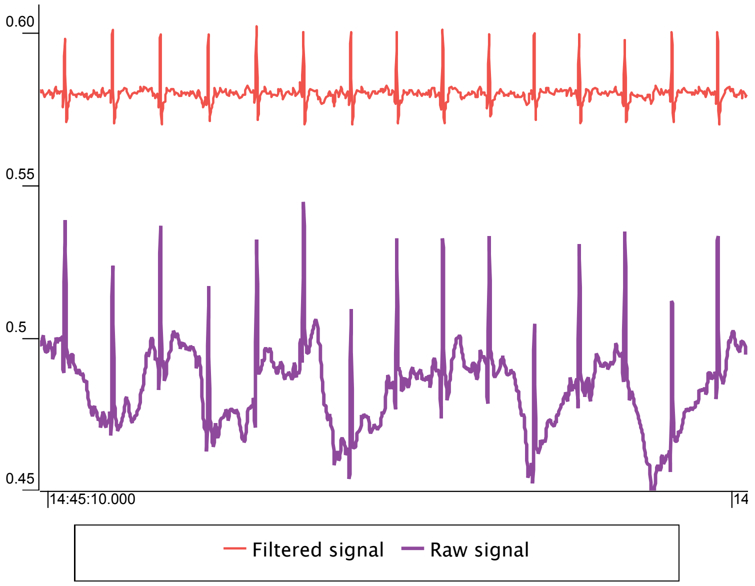


Figure 6.3 – ECG filtering using an adaptive filter fed with the accelerometer data

- Erosion: This operator takes as argument a signal and a structuring element. It will convolve the structuring element onto the signal and output the area that has been covered by the center of the structuring element.

- Dilation: This operator also takes as argument a signal and a structuring element. But during the convolution it will output the area covered by the structuring element.

An example of the result of these operators is presented in Figure 6.4.

However, MM can also be used with temporal signals, and in our context, ECG signal. In [82] the author proposes a definition of the MM operators for multidimensional signals. In his paper, Yazdani presents a simpler definition and focuses on the operators required for QRS extract. The operators are listed in Equation 6.2, with $g(n)$ is the structuring element of length n use by the operators, and f is the signal that we want to alter.

$$\begin{aligned} Dilation\oplus : f \oplus g(n) &= \max(1 \leq i \leq n)\{f(i) + g(x-i)\} \\ Erosion\ominus : f \ominus g(n) &= \max(1 \leq i \leq n)\{f(i) - g(x-i)\} \\ Open\circ : f \circ g(n) &= (f \oplus g) \ominus g(n) \\ Close\bullet : f \bullet g(n) &= (f \ominus g) \oplus g(n) \\ Top-Hat : THat(f, g(n)) &= f(n) - f \circ g(n) \\ Bottom-Hat : BHat(f, g(n)) &= f(n) - f \bullet g(n) \end{aligned} \tag{6.2}$$

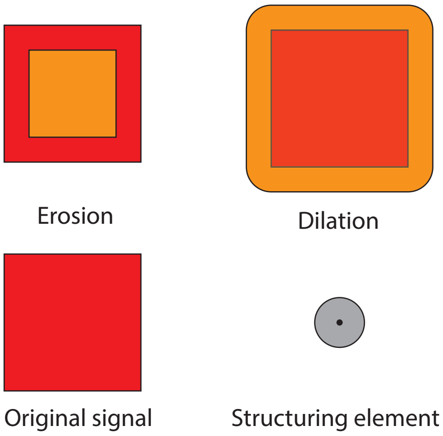


Figure 6.4 – MM core operators

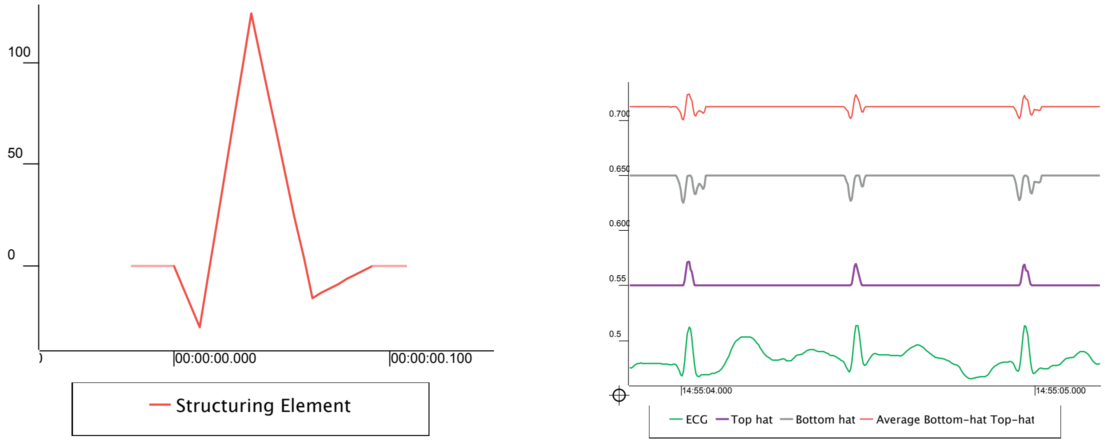
6.3.3 QRS extraction

The approach taken by Yazdani to extract QRS starts by applying a low-pass filter with a cut frequency of 50Hz to the ECG signal in order to limit the impact of frequencies which could not have been generated by the heart electrical activity. This filtering is implemented using a Butterworth filter that provides a flat response in its pass band, allowing to effectively cut higher unwanted frequencies while minimising the distortion of the signal.

The rest of the QRS extraction algorithm relies on mathematical morphology: The structuring element used by the MM operators represents coarsely a QRS complex (Figure 6.5a). It is composed of 5 points: the standard Q, R and S points, and the onset and offset of the complex. These last two points represent the extremities of the QRS complex and, similarly, the extremities of the structuring element.

The ECG signal is processed in small chunks of 2 seconds. This mechanism allow for a better adaptability. If a QRS is successfully detected during these 2 seconds, the algorithm will update its structuring element and will restart the processing starting from the last successfully detected QRS. If no QRS complex were detected, it will proceed with the next chunk of 2 seconds.

When processing a chunk of the signal, the algorithm applies two mathematical morphology transforms: top-hat and bottom-hat. These transforms enhance peaks and valleys in the signal and remain flat otherwise, as depicted in Figure 6.5b. The average of the two transformed signals ideally yields non zero signal for the QRS complex of heart beats, and zero outside of these complexes (The red signal in Figure 6.5b). More specifically, the averaged signal follows a rough representation of the QRS and exhibits a peak at the R-wave location, and valleys for Q and S locations. However, with real life ECG signals, artefacts appear due to noise. These artefacts are handled by applying heuristics based on the temporal limitations of human QRS complexes:



(a) Structuring element

(b) MM operators on ECG signal (y-axis normalised for display)

Figure 6.5 – Structuring element and mathematical morphology operations on ECG

- A QRS segment cannot be shorter than 17ms. This represent the time between the Q point and the S point.
- The difference between two consecutive heartbeats cannot be shorter than 250ms. This difference is computed between the R peaks of the respective beats. This limit can be interpreted in term of heart rate, as beats separated by 250ms would give a heart of 240 beats per minutes.
- the last heuristic states that the difference between two consecutive heartbeats cannot be longer than 1800ms. This limit is also interpretable as a heartrate of 33 beats per minutes.

To better adapt this method to different users, the structuring element is updated with each new QRS complex detected. This update is performed for each of the 5 points representing the QRS complex, both in term of temporal update and amplitude update by applying the formula:

$$\begin{aligned} NewTime &= (1-\alpha) \times CurrentTime + \alpha \times ExtractedTime \\ NewAmp &= (1-\alpha) \times CurrentAmplitude + \alpha \times ExtractedAmplitude \end{aligned} \tag{6.3}$$

This formula is parametrised by α which determines the update rate. In order for the algorithm to converge toward the best representation of a QRS complex for the structuring element, the α is adapted based on a $Peak_activity$ value which can be computed as the sum of the absolute values of the QRS complex. α is initialised as 0.9. If consecutive $Peak_activity$ (corresponding to the detection of 2 consecutive QRS complexes) are too different from each other the learning rate is decremented by 0.05. Conversely, if consecutive $Peak_activity$ are similar, the learning rate α is incremented by 0.05.

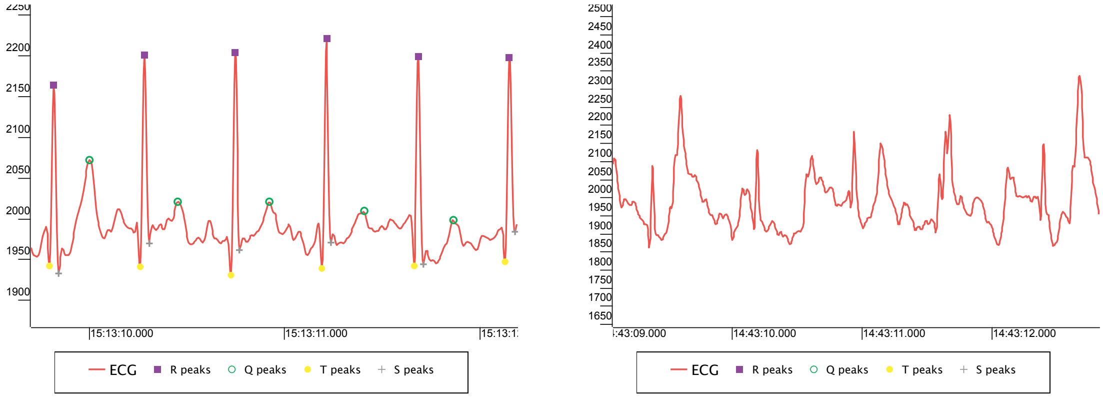
While QRS complexes are relatively easy to locate on an ECG, P and T points are less obvious, and may even be missing. We do not consider P points since according to [71], they carry little information about the state of glycaemia of the users. On the other hand, the T points are instrumental in discriminating the different glucose states. They are therefore extracted using a gradient ascent after each QRS complex since T points corresponds to the first of the two maximums between two consecutive heartbeats (the second maximum corresponding to the P point of the next heartbeat if present).

However, this method has its limits, as noise can still be present despite the signal prefiltering. While the algorithm used to detect the QRS complex can still detect QRS complexes under high noise, the distortion in the amplitude of the complexes make them unreliable to detect the subtle variations we are interested in. To prevent the use of “corrupted” heart beats that are detected but that are too noisy to be exploitable for glycaemic state detection, we compute a sliding approximate entropy [92], and we label the complexes with high entropy as being unsuitable for the evaluation of the glycemic state (However, the QRS is still extracted by the algorithm in order to compute the heart rate of the user). A distinction between segments of high and low entropy is presented in Figure 6.6. The sliding approximate entropy uses the standard notion of entropy. Low entropy represents an understanding of predictability while high entropy is highly unpredictable. The approximate entropy allows to measure this entropy on slices of time series data. It is controlled by two parameters m representing the amount of data being compared, and r representing the level of filtering. Given a time series $\mathcal{T} = [x_1, x_2, ...x_N]$, and parameter m and r, the algorithm proceeds as follow:

1. For $i = 1 toN - m + 1$, build the vectors $v_i = [x_i, x_{i+1}, ..., x_{i+m-1}]$
2. For $i = 1 toN - m + 1$, build the numbers $C_i = \frac{\text{number of vectors } v_j \text{ such that } d(v_i,v_j)\leq r}{N-m+1}$ with $d(v_i, v_j) = \max a|v_i[a] - v_j[a]|$
3. Compute $h(m) = (N - m + 1)^{-1} \sum_{i=1}^{N-m+1} log(C_i)$
4. Approximate entropy is given by $H = h(m) - h(m + 1)$

In the specific context of d1namo and its dataset, we use the following parameters: $m = 2$, and $r = 80$. We empirically defined the threshold above which the signal is discarded by $T_{entropy} = 0.09$.

Once the QRS complexes are extracted, the discriminating features presented in Table 6.1 are computed for each QRS. The three first features (R amplitude, T amplitude and ST elevation) can be read directly in the QRS complex. The QT c-Fridericia necessitates the heart rate of the patient. This heart rate is computed as the number of R peaks detected per minutes by using a sliding window of 1 minute. Finally, T area and R area are computed as the integrals of the ECG between respectively the Q point and S point for and between the QRS offset and the next time the signal reaches the offset amplitude once it has passed the T wave.



(a) Low entropy ECG

(b) High entropy ECG

Figure 6.6 – Comparison of the QRS detection between low and high entropy Signal

6.4 Evaluation

6.4.1 Dataset collection

We evaluate our approach on data from 9 diabetes type 1 patients. This data was collected during a campaign sanctioned by the ethical committee of Canton de Vaud, Switzerland. Each participant was asked to wear a Zephyr bioharness sport belt for 4 consecutive days. The belt was collecting three main signals: *electrocardiogram (ECG)*, *tri-axial accelerometer*, *breathing sensor readings*. These signals have respective sampling rates of 250Hz,100Hz, 25Hz. The battery of the belt lasts around 24 hours. Therefore, the participants were wearing it during day time and charge the belt during the night. The participants were also asked to wear the Enlite sensor of the Ipro 2 from Medtronic which is a professional CGMS that measure the glycemic level every 5 minutes. In addition, for calibration purposes, the CGMS requires manual glucose monitoring at a few hours interval, leading to a set of glucose values that were measure directly through a glucometer.

Since the goal of D1namo is to measure the glucose level in real life condition, the participants were not given any specific activity to perform or diet to follow, but were instructed to follow their normal daily routine.

Additionally, the participants were asked to log their food intake, as well as the units of insulin they took. The participants were asked to take a picture of all the food, snacks and drinks they were having next to a coin of 1 Swiss franc in order to have a scale of reference. In case the participants could not or forgot to take pictures, they were asked to provide a description of their meal as detailed as possible. This method of data collection is not systematic and relies on the thoroughness of each user. Furthermore, it introduces noise due to the variations in the composition of similar meals. For example the proportion when cooking homemade meals may vary between different versions of the same dish. Although imposing a strict diet to

Number of participants	9
Age average	40 years
Weight average	72 kg
Percentage male participants	66%
Average food logs per participant per day	4.25
Days per participant	4
Total days exploitable	29
Total number hypoglycaemic events	28

Table 6.2 – Statistics on the data set collected

the participants or providing them with glucose solutions in a controlled environment would ensure the accuracy of the glucose intake logs, these approaches were incompatible with the goal of D1namo to infer glucose level in real life conditions.

The insulin logs were also provided manually by the participants and contained *time of injection*, *brand of the insulin injected*, *number of units injected*. Diabetes type-1 patients are used to log their insulin intakes thoroughly and the precision with which the quantities of insulin are measured before injection makes it a reliable source of information.

Due to the lack of glycaemic data for certain days or belt not properly worn, 7 days were excluded from the evaluation. The resulting statistics about the dataset are presented in Table 6.2.

CGMS preprocessing The CGMS taps into interstitial fluids in order evaluate the blood glucose level of the user. However, the measured value from the CGMS has a delay compared to the true blood glucose level. This delay is due in part to interstitial fluid glucose that is physiologically delayed compared to blood glucose as described in [100], but can also be due to the measuring device itself. The lag is of the order of 10 minutes, and is subtracted to the CGMS readings of each user. We define the value of the lag for each user as the value between 5-15 minutes that minimises the error between the CGMS and the manual glucose values.

6.4.2 Physiological model

The physiological model is evaluated on a classification test. Each heartbeat is evaluated. Conceptually, heartbeats are defined by the features described in Table 6.1, as well as one of the classes *Hypoglycaemia* or *Euglycaemia*: whether the beat happened during a time when the user had a low blood sugar or a normal one. The threshold to determine if a blood sugar value is low is defined according to our standard definition of hypoglycaemia: 3.9 mmol.l^{-1}.

In order for contiguous beats to not belong to training and testing sets, we separate the data into days, and perform a leave-one-out cross validation on the days. Which means that given

4 days of data, we train the model using 3 days and test it with the remaining day and we repeat this evaluation for each of the 4 days. Furthermore, given the scarcity and duration of hypoglycaemic events, we perform classes virtual balancing, in order to avoid biases. While this method does not add (over-sampling) nor remove (under-sampling) samples to the dataset, it ensure that the algorithm will not favour the majority class when optimising its objective function by attributing a weight to each class in order to maintain similar weighted classes.

We use the standard machine learning library Weka [59], and we consider 3 different algorithms: decision tree C4.5 with pruning, random forest and logistic regression. Once the classification performed, we average the results over a minute. Indeed, blood glucose level has a momentum which can be leveraged to prevent a heart beat to be classified as *Euglycemia* if it is surrounded by 2 beats classified as *Hypoglycaemia*.

The algorithms are evaluated in terms of accuracy to get a general appreciation of their performance, as well as specificity and sensitivity for the class *Hypoglycemia* which is the class of interest.

6.4.3 Comparison with existing approaches

We compare our approach against the results obtained by two state-of-the-art algorithms presented by San and Lin in [105] and [77] respectively. San proposes a framework based on deep belief network. This framework uses the heart rate of the user, as well as the QTc-Fridericia to first train a single layer restricted Boltzmann machine to learn a feature representation in an unsupervised manner. This representation is then used as input for a feed-forward neural network (FFNN) to perform the classification of the heartbeat as having been produce during hypoglycaemia or not. This approach is completely unsupervised.

The second approach taken by Ling is based on the concept of extreme learning machine. This concept uses shallow FFNN with only 1 hidden layer and proposes a different learning scheme from standard FFNN. Indeed, the weights of the edges linking the input node to the hidden layer are initialised at random and never update. On the other hand, the weight linking the hidden layer to the output are training in a single step. This technique has the advantage of decreasing the learning time of the model while proposing interesting performance. The extreme learning machine network proposed by Ling receives 4 inputs: heart rate, QTc-Fridericia, Δ heart rate and Δ QTc-Fridericia. The Δ features being computed between heartbeats to capture the respective variations of heart rate and QTc-Fridericia.

Both approaches have been evaluated with the same dataset of 16 diabetic type-1 children who stayed 10 hours overnight at an hospital to collect data, representing the collection of 589 samples. This setting is different that the real life approach taken by D1namo. Hospital-grade measurements are expected to be of better quality than the one collected by the sensor belt in D1namo.

Algorithm	Sensitivity	Specificity	accuracy
Logistic regression	0.79	0.65	0.67
Random forest	0.68	0.58	0.60
Decision tree	0.68	0.63	0.64
DBN [105]	0.80	0.50	-
ELM-NN [77]	0.78	0.60	-

Table 6.3 – Evaluation of the physiological model

6.4.4 Results

The results of our evaluation are provided in Table 6.3. The three models generated by D1namo are put in perspective with the two approaches presented before, and that are considered the state-of-the-art. While the random forest and the decision tree performance are below the other approaches, The logistic regression yields the best results for the three metrics considered. This approach has on par performance with the state of the art although our approach handles more noisy data collected in an every day life conditions, compared to ECG captured overnight in an hospital setting. These results show that the approach taken by the physiological model is adequate. However, we also need to evaluate the runtime performances of our approach to ensure that it is suitable for mobile computation.

6.4.5 Performance evaluation

To complete our evaluation, we test the runtime of our algorithms on mobile device. We use an Android smartphone Nexus 5, with a quadcore CPU of 2.3GHz, 2GB of RAM with the operating system Android 6.0.1. We first evaluate the features extraction algorithm which includes the adaptive filtering of the ECG signal, the detection of the fiducial points using the mathematical morphology approach as well as the computation of the features presented in Table 6.1. The results are presented in Figure 6.7. We also display on the figure the line y=x corresponding to a processing time as long as the data acquisition itself. Processing time above this line would not be sustainable as the algorithm would not be able to keep up with the stream of data. However, the fact that we use incremental algorithm allow processing time to stay below this line. Following a similar cycle than a CGMS, we propose a duty cycle every 5 minutes. According to Figure 6.7, the smartphone will require only 70 seconds of computation to extract the ECG features.

The classifying task is tested using a Weka implementation for mobile ([102]). It consists of the standard Weka library stripped from its graphical components. We focus on the logistic regression algorithm which yields the best results. The classifying task itself is extremely fast and a day worth of data can be processed in a bit more than a second (average of 1130ms). However building the classifier is more demanding and can be performed under 45 seconds for a day worth of data. While this number is very low, logistic regression is by default not

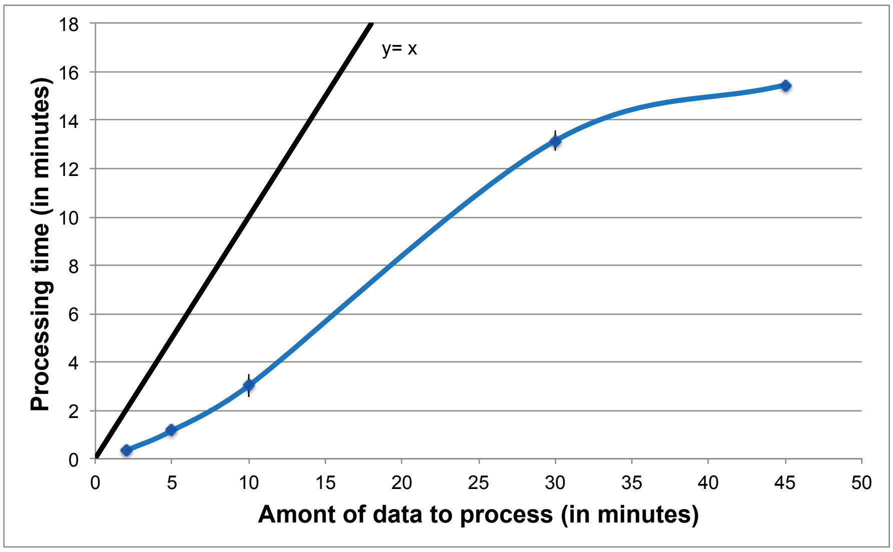


Figure 6.7 – Processing time as a function of the length of the signal to process

incremental, which means that the model needs to be trained on the full dataset each time and which is not suitable. However, this model can be trained offline using initial data of the patient and then downloaded to the phone to only perform the classifying task.

6.5 Conclusion

In this chapter, we presented a physiological approach to hypoglycaemia detection. This approach leverages incremental algorithms to process signals coming from a sport belt. This processing involves multi phases of filtering to address the noisy nature of the collected signal: A first adaptive filters alleviate the noise of the ECG by using its correlation the movements of the user, captured by the accelerometers. A QRS extraction algorithm, based on mathematical morphology operators is then used to segment the ECG signal into heart beats and to detect the location of fiducial points of this heart beat. Then, we extract the features relevant for the detection of hypoglycaemia as described in by Laitinen and reproduced in Table 6.1. Lastly, an approximate entropy is computed as a safeguard to prevent the use of heart beats which are too deteriorated to be used for hypoglycaemia classification.

We finally study the performances of our approach both in term accuracy and mobile performances. We found that our approach have similar performances compared to the state-of-the-art while handling very noisy data. Furthermore, the use of incremental algorithm makes it possible to legitimate the use of our approach in a mobile environment. The low run time necessary to filter and processed the ECG signal on a smartphone confirmed this possibility.

7 Energy model

7.1 Introduction

Because diabetic type-1 patients do not produce insulin naturally, they need to inject it from external sources (e.g. manual injections or insulin pump). In order to decide the correct amount of insulin to inject, a diabetic patient needs to take into account a high number of parameters. Among these parameters we can list her current glucose level and its recent evolution, the amount of insulin recently injected, the amount of food that she had recently and the amount that she plans to have in a near future and the composition of this food. The patient also needs to account for the physical activities she will perform.

All these variables can be grouped in two categories that need to be balanced. On one hand the glucose intake which tends to increase the blood glucose level. On the other hand, the glucose consumption that uses or stores the available glucose. Based on this observation we propose an energy model that evaluates both categories in order to continuously estimate the blood glucose level of the patient.

7.2 Energy model

An overview of the model, in Figure 7.1 reveals that it is based on the balance between 3 quantities:

- The energy expenditure that is “burning” the glucose available in the bloodstream.
- The glucose coming from the food intake.
- The insulin intake, which can be seen as reducing the glucose in the blood.

This model overlooks several factors that impact blood glucose level, such as the transformation and use of the glucose in the body as well as the impact of other hormones than insulin

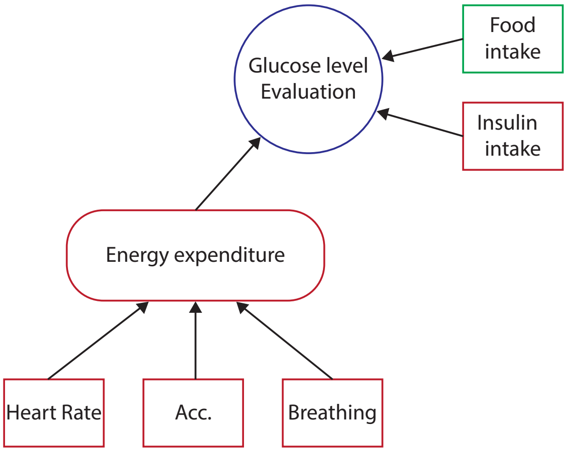


Figure 7.1 – Overview of the energy model

such as glucagon, but it nonetheless covers the three essential sources and sinks of glucose. This model is in some sense similar to the one presented by Dalla Man et al. [35], in the sense that it combines energy intake, insulin level and physical activity. However, our model remains more simple, intuitive and relies on non-invasive observations. In the rest of the section, we will go through these three quantities and see how they can be compared.

7.2.1 Evaluation of the energy expenditure

The glucose intake of a user can be roughly estimated using the nutrition information of her meal, the insulin intake is precisely measured when injected. However, the energy expenditure is harder to estimate, whether it be for physical activities such as biking, household activities, or for the basal metabolic expenditure (i.e. the standard energy expenditure that comes from the body simply functioning). Instead of focusing on the type of activity that the patient is performing, we propose an energy expenditure model representing the intensity of the effort performed by the user through the day. While this model does not allow to characterise two activities of the same intensity, it can understand the difference of burned calories between an intense or more relaxed version of the same activity. As depicted in the overview figure, we leverage the three different sensors available on the Zephyr Bioharness belt to define the energy expenditure.

The accelerometer is commonly used to measure the level of activity of the user through the VMU (vector magnitude unit) of all its axis. However, there are cases where accelerometers can be deceived, for instance when the user is using some mean of transportation, or when she's at the gym.

The heart rate (HR) is also a good indicator of the activity level. Indeed the level of activity is related to the effort produced by the muscles of the user. When producing this effort, the

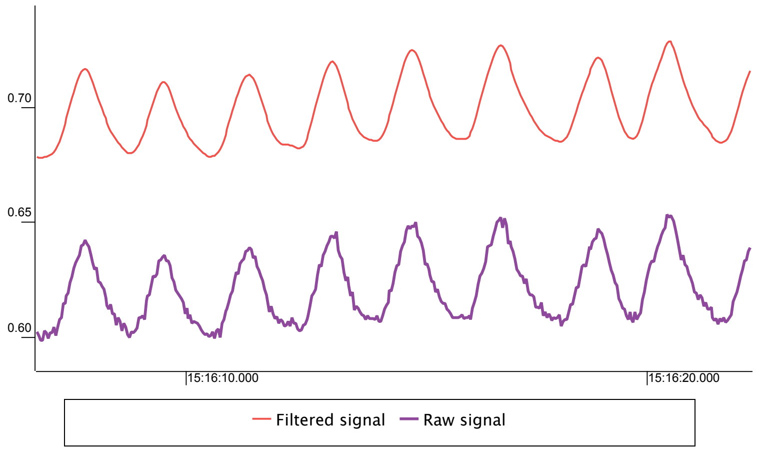


Figure 7.2 – Breathing filtering using a low-pass filter

muscles require energy and oxygen. In order to provide enough oxygen to the muscles during intense activity, the heart needs to pump blood faster, therefore increasing the heart rate.

Finally, **the breathing rate** (BR) is also an important indicator of the activity level, for the same reason than the hear rate: to produce an effort, the muscles require an intake of oxygen.

However, BR and HR are also impacted by other factors, such as stress, or strong emotions in general. Therefore, we rely on the combination of the three sensors in order to compute the activity level of the user. We first extract these three quantities and then combine them to produce a measure of energy expenditure as accurate as possible.

Preprocessing

In order to cope with the inherently noisy data from the belt, we included a preprocessing phase to improve the raw signals. The breathing data contains white noise. However, this noise has a higher frequency than the breathing rate, which stays under a 3 hertz. Therefore, a low-pass filter with a cut frequency of 3Hz allows for a clean signal from which we can extract the breathing rate. This filtering is done using a Butterworth low-pass filter.

The ECG data is preprocessed by following a similar pipeline to the one presented in Chapter 6. This pipeline includes: 1/ Apply an adaptive NLMS filter on the ECG signal using the accelerometers as reference signals, 2/ perform the QRS complexes extraction by applying the mathematical morphology approach. Since we are only interested in extracting the heart rate from the ECG, it is not necessary to compute the approximate entropy as the QRS algorithm is reliable at detecting QRS complexes even under noise.

We do not perform any filtering on the accelerometer signal.

Features extraction

Once the signals cleaned from the noise, we extract the features of interest. The vector magnitude unit (VMU), is defined as the Euclidian distance along all the axis of the accelerometer.

$$VMU = \sqrt{Acc_{vertical}^2 + Acc_{lateral}^2 + Acc_{sagittal}^2}$$

The VMU is computed and averaged using a sliding window of 1 min in order to smooth it.

The heart rate is computed from data derived from the ECG as described in Chapter 6, Section 6.2). From the fiducial points extracted, we define a heartbeat at the location of the R peak. The extracted R peaks are counted and averaged over a sliding window of 1 minute in order to define the HR of the user.

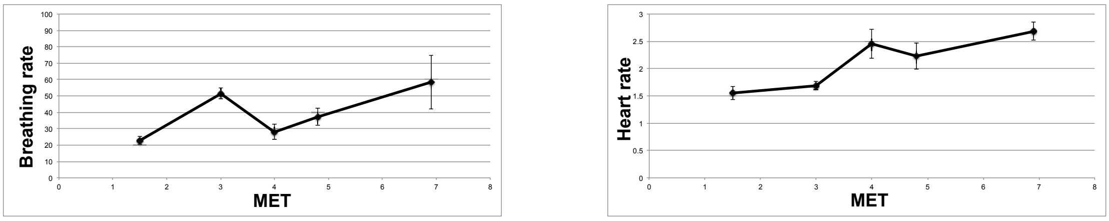
Finally, the breathing rate is computed by extracting the local maximums on the breathing signal. These maximum are found by traversing the time series while keeping track of its derivative. A change from positive derivative to negative indicate a local maximum. It can be seen from Figure 7.2 that these maximums correspond to the maximum inhalation and serve as a reference for a breath cycle (inhalation/exhalation). Once the peaks extracted, another sliding window of 1 minute to alleviate sharp changes in the BR of the patient.

Definition of the activity level

The level of activity is computed in metabolic equivalent tasks (MET) which is a measure of the energy spent by the user with regards to her basal metabolic rate (BMR), which is the minimum amount of energy spent by a human at rest. The MET of a large amount of activities has been previously defined in a compendium of physical activities [6]. We use the values of the compendium as a reference to calibrate the detection of physical activity.

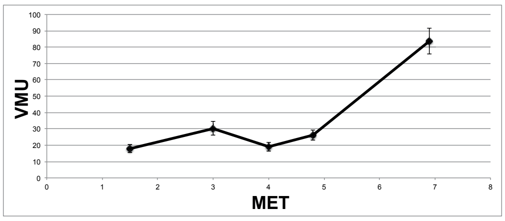
For each signal VMU, HR and BR, we want to define the MET of the patient at each time t as a function of this signal . In order to establish the parameters of these functions, a ground truth is required to be able to use the compendium. This ground truth is built using the signals of 3 participants who performed 5 tasks (walking at moderate speed, running at 5mph, bicycling <10mph, sitting, climbing stairs). From these activities, we built a model of the evolution of the HR, BR, and VMU as a function of MET, as depicted in Figure 7.3. We perform a linear regression for each of the signals, these regressions become the models to determine the MET of the patients based on their signals.

Once the 3 activity levels (corresponding to the 3 different sensors) are computed, they need to be aggregated in order to have a unique value of the MET. We previously discussed cases where the signals HR, BR and VMU are altered by phenomenon independent from physical activities and therefore do not accurately represent the actual activity level of the user. Since



(a) Breathing rate

(b) Heart rate



(c) Vector Magnitude Unit

Figure 7.3 – Evolution of the signals HR, BR, VMU as a function of the activity (in MET)

Activity	Sitting	walking	biking	stairs climbing	jogging
MET value [6]	1.5	3	4	4.8	6.9
RMSE	0.21	0.54	0.51	0.74	0.61

Table 7.1 – Root mean square error of the MET computation from averaged linear regressions

we have three variations of the MET available, we propose an aggregation scheme that takes into account the two closest METs. This scheme is particularly elegant since, if all three activity levels are accurate, discarding one of them will not change the outcome, while if one the levels is an outlier, discarding it will allow to compute an accurate activity level. For instance, if the heart of the user is beating faster than normal because of the stress, the model based on HR will yield a high MET whereas the two other models will have lower MET. Therefore, the MET value of the HR will be discarded for this specific time and the energy level will be computed as the average of the MET coming from the breathing rate and the one coming from the VMU. Once the MET is evaluated it can be converted to calories based on the formula presented in [66]:

$$\text{Energy expenditure (calories.minutes}^{-1}\text{)} = \frac{\text{MET} \times 3.5 \times \text{weight(kg)}}{200}$$

While we could not evaluate the accuracy of the prediction of the level of activity on the patients due to the lack of information regarding their daily activities, we tested the regressions against the three participants for which we had a ground truth, to show that their values are consistent. The root mean square error of the MET determined using the averaged regression is given in table 7.1. The resulting MET for the various activities is displayed on figure 7.4.

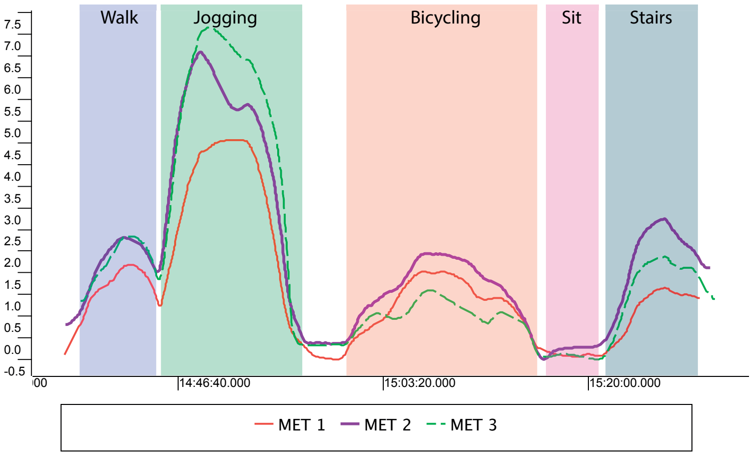
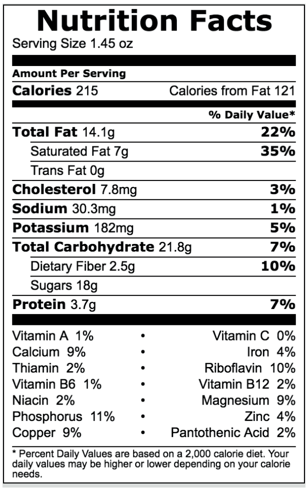


Figure 7.4 – MET of the 3 participants under various activities



Nutrition Facts

Serving Size 1.45 oz

Amount Per Serving	
Calories 215	Calories from Fat 121
	% Daily Value*
Total Fat 14.1g	**22%**
Saturated Fat 7g	**35%**
Trans Fat 0g	
Cholesterol 7.8mg	**3%**
Sodium 30.3mg	**1%**
Potassium 182mg	**5%**
Total Carbohydrate 21.8g	**7%**
Dietary Fiber 2.5g	**10%**
Sugars 18g	
Protein 3.7g	**7%**

Vitamin A 1%	•	Vitamin C 0%
Calcium 9%	•	Iron 4%
Thiamin 2%	•	Riboflavin 10%
Vitamin B6 1%	•	Vitamin B12 2%
Niacin 2%	•	Magnesium 9%
Phosphorus 11%	•	Zinc 4%
Copper 9%	•	Pantothenic Acid 2%

* Percent Daily Values are based on a 2,000 calorie diet. Your daily values may be higher or lower depending on your calorie needs.

Figure 7.5 – Nutritional information as can be seen on fitbit

7.2.2 Food intake modeling

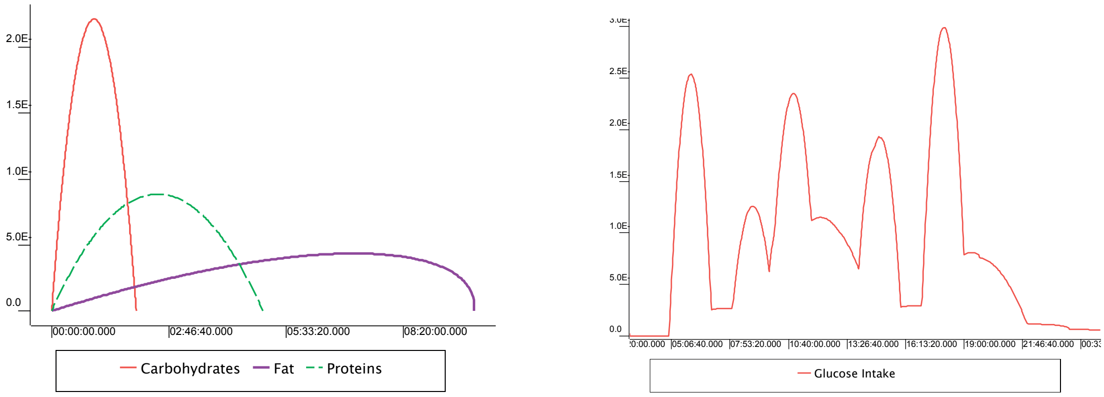
The food intake model is based on the annotations of users and the pictures they take of their meals and various snacks. In order to estimate the amount of glucose the food will provide once digested, we need its composition. We use online food databases such as *Fitbit food* [1] and *my fitness pal* [2] to get the composition of the different meals (cf. Figure 7.5) based on the information provided by the user. These services provide the nutrition information for a wide variety of raw and processed food, also tapping in brands database to provide the nutrition information of specific products. They can be accessed from a browser or directly from a web-service.

We focus the model essentially on the carbohydrates, fats and protein incomes, as they are

---

[1] http://www.fitbit.com/foods

[2] http://www.myfitnesspal.com/food



(a) Glucose conversion rate

(b) Glucose intake for a day

Figure 7.6 – Conversion of the different components into glucose over time

the different sources of glucose once digested. They are released over various times through the digestion. Carbohydrates will be the fastest to be digested and are transformed into glucose within 2 hours from ingestion, they represent the main source of glucose. Simple carbohydrates are easily broken into glucose while more complex ones take more time, finally carbohydrates such as fibres are not digested. It is estimated that 90%-100% of the carbohydrates are converted into glucose. Proteins are slower to process and its digestion can last for up to 5 hours after the meal. They can to some extends be metabolised and reduced to glucose via gluconeogenesis which is the generation of glucose from non carbohydrate compounds. In this thesis, we consider that 50% of the proteins are converted into glucose. Finally, fats are the slowest to be processed (10 hours), and 10% of the fat ingested will be converted to glucose. The different speeds of conversation can be compared in Figure 7.6a, and an example of the evolution of the glucose income over a day is displayed in Figure 7.6b. Using these pieces of information, it is possible to build a distribution over time of the glucose provided by the food at time t, as shown in Figure 7.6b. The food intake is then augmented by a scaling factor. This factor is estimated using grid search to minimise the error of the general model. The glucose available in the blood stream is then computed as the integral of this distribution.

We use the family of beta distributions in order to model the conversion rate of each food components. These distributions are particularly convenient as their probability distribution is defined between $[0,1]$. The distributions are defined by two parameters $0 < \alpha$ and $0 < \beta$. depending on these parameters the resulting distribution will exhibit different shape property. The probability density function of a beta distribution can be computed using Formula 7.1 where $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ and which can be approximated using the Stirling approximation $\Gamma(z) \approx \sqrt{2\pi z} \frac{z}{e}^z$. The parameters used to generate the distributions presented in Figure 7.6a are:

- Carbohydrate distribution: $\alpha = 2.0$, $\beta = 2.0$
- Proteins distribution: $\alpha = 2.0$, $\beta = 2.0$

- Fat distribution: $\alpha = 2.0$, $\beta = 1.4$

$$pdf(\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{7.1}$$

Once the distributions computed, they are stretch in time in order to cover the digestion time described above. In order to conserve the properties of a probability distribution, the distributions are also rescaled to ensure that their integral sums up to 1. Finally, the distributions are again rescaled to take into account that only a percentage of each components is transformed into glucose. In this way, to model the ingestion of 5 grams of sugar at time t, the system only needs to translate the corresponding distribution in Figure 7.6a in time by t, and multiply it by 5. For each food intake the distribution are accumulated leading to the distribution of the glucose intake for a day presented in Figure 7.6b.

7.2.3 Insulin intake modeling

In a similar fashion, we model the insulin action in the blood stream, from the logs of insulin intake written by the participants. Based on the brand of the insulin injected, we infer its type (fast acting, short acting or long acting). The insulin is taken in international units (u), which is defined as the biological equivalent of 34.7 micrograms of crystalline insulin. The amount of glucose that can be "neutralised" [3] by a unit of insulin is varying between persons and depends on a correction factor giving the number of mmol.l^{-1} of glucose that a unit of insulin can "counter":

$$\text{Correction factor} = \frac{100}{\text{total daily dose of insulin}}$$

We therefore multiply our model by this correction factor in order to build the distribution over time of the amount of glucose that can be countered by insulin. Similarly to the food intake, the release of the insulin in the bloodstream varies depending on the type of insulin. We model the amount of insulin available in the bloodstream in a similar fashion to the food intake, using the following parameters for the beta distribution:

- Fast acting insulin distribution: $\alpha = 2.0$, $\beta = 4.0$
- Intermediate acting insulin distribution: $\alpha = 1.4$, $\beta = 1.3$
- Long acting insulin distribution: $\alpha = 2.5$, $\beta = 5.0$

---

[3] Formally, insulin is the hormone responsible for triggering the transformation of glucose into glycogen in order to be stored for later use.

7.2.4 Aggregation

Lastly, in order to be able to compare the three quantities (glucose intake, insulin intake and energy expenditure) together, we need to harmonise them to the same unit. The insulin intake is already converted to mmol.l^{-1} of glucose that can be canceled. Therefore, we need to also convert the energy expenditure and the glucose intake into the same unit. Glucose intake is in grams of glucose, we convert it to mmol using the molar mass of glucose (180.1559 g.mol^{-1}). We then normalise this quantity by the average number of litre of blood in an adult human body (5l). The energy expenditure is given is calories. We convert the model into grams of glucose using the fact that a gram of glucose contains 3.75 calories. From there the same conversion used for the glucose intake is applied. The quantities are then combine as the following:

$$\text{Glucose level (mmol.l}^{-1}\text{)} = \alpha(\text{Glucose intake} - \text{insulin intake} - \text{energy expenditure}) + \beta$$

As stated previously, the model is very simple and can overlook some aspects. For instance, the correction factor defining the impact of the insulin on the blood glucose is very coarse, and we do not take into account the energy required for the digestions of the different sources of glucose, or the impact of other hormones than insulin. To alleviate these problems and improve the accuracy of the model, each user is given a scaling (α) and additive (β) factors which accounts for discrepancies in the model due to an over generalisation and lack of data at nights. These factors are defined for each user and do not affect the evaluation as they stay the same across the days of each user. These factors are optimised through a grid search in order to minimise the model's error.

7.3 Evaluation

We evaluate our approach on the same dataset presented in the previous chapter of 9 diabetes type 1 patients. The energy expenditure model yields a representation of the blood glucose level. It can therefore be compared directly to the ground truth. We computed the root mean square error (RMSE) between the models established and the ground truth from the time-advanced continuous glucose monitor (CGMS). We also compute the RMSE between the glucose evolution provided by the CGMS and the manual glucose samples taken regularly (on average 25 measurements per participants) using a standard glucometer. These measurements are considered the most accurate and are used to calibrate the CGMS. The results are presented in Table 7.2.

Our model displays a relatively high RMSE on average when compared to the CGMS. However, this is parly due to outliers as it can be seen through the very high maximum RMSE. While the energy expenditure is automatically measured in our system, the food is manually logged by the user. This is a tedious and error-prone task. Variations in the food intake can considerably affect the performance of our algorithm. Figure 7.7 shows that days when the participants did not have many log entries of their food (typically, days with 1 food entry for the entire day)

Comparison	RMSE	Std dev	min	max
Model - CGMS	4.80	3.15	1.55	15.34
Model - Glucometer	4.34	4.15	0.34	20.7
CGMS - Glucometer	2.22	2.23	0.59	7.10

Table 7.2 – Pairwise RMSE between our model, the CGMS readings and manual glucometer measurements

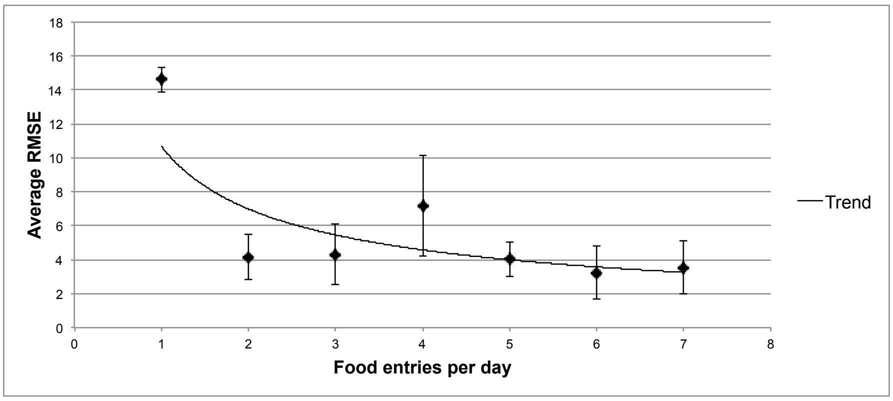


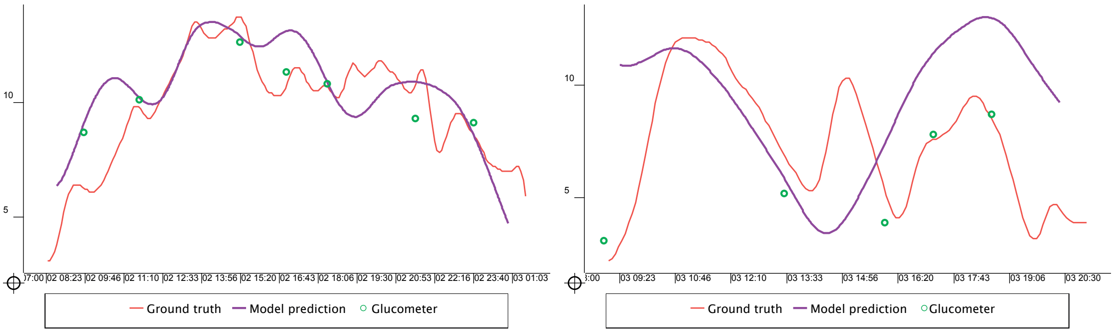
Figure 7.7 – Evolution of RMSE as a function of the number of entries per day

led to very high RMSE. This can be explained by having elevations in the blood glucose level captured by the CGMS that cannot be found in our model due to missing data (c.f. Figure 7.8b).

While the CGMS remains more precise than our model, we can see that the CGMS itself is not perfectly matching the measurements coming from the glucometer. This is explained by the fact that the CGMS measures the glucose level in the interstitial fluids and therefore does not offer a direct observation of the blood glucose level. Another consequence of this type of measurements is a lag between the glucose level from the blood and from the interstitial fluids. Finally, the CGMS output is continuously calibrated by the glucometer readings, therefore theoretically reducing the error between the two values.

7.3.1 Performance evaluation

To complete our evaluation, we test the runtime of our algorithms on mobile device. We use an Android smartphone Nexus 5, with a quadcore CPU of 2.3GHz, 2GB of RAM with the operating system Android 6.0.1. The energy and insulin intake are sparsely distributed during the day and are therefore not a computational bottleneck in our approach. The energy expenditure evaluation on the other hand follows a complex processing pipeline. We therefore focus our performance evaluation on this part of the system. We perform our activity detection on time range of data from 2 minutes to 45 minutes. We can see in Figure 7.9 that the total computation time taken to estimate the energy expenditure of the user is dominated by the computation of the heart rate. This is due in part to having a higher sampling rate for the ECG



(a) model of a day with low RMSE (in mmol.l^{-1})
(b) Model of a day with high RMSE (in mmol.l^{-1})

Figure 7.8 – Variations in the accuracy of the model

signal than accelerometers and breathing signal and also because the heart rate is derived from the fiducial points extracted for the physiological approach.

7.4 Conclusion

In this chapter, we presented our approach to evaluate the glycemic state of the patient based on a non-invasive energy model. This model is based on energy expenditure, energy intake, and insulin. Following the hierarchical approach to complex events detection, the model first evaluates the energy expenditure based on 3 sources that are the breathing rate, the heart rate and the accelerometer. This energy expenditure is then used in conjunction with the food and insulin intake of the patient to evaluate a more complex information: The blood glucose level of the patient.

The energy expenditure that we present deviates from the standard approaches as it does not focus on the detection of specific activities, it rather yields the intensity of the physical activity performed by the patients. This measure is more relevant for diabetes patients as it allows an accurate evaluation of the energy burned by the patient.

The evaluation of the energy model shows encouraging results. However, this model relies essentially on available observations and is therefore limited. It does not account for other factors impacting the glycemic level such as the atypical endogenous production of glucose by diabetic patients, the dynamics of insulin resistance when performing physical activity or the impact of other hormones. It rather follows the intuitive approach taken by diabetics and diabetologists who evaluate the amount of insulin to inject based on their current glucose level, their food and previous insulin intakes and their physical activity. While the general trends of the glucose variations can be captured by our approach, the extension of the model with the additional factors listed above should reduce its error and allow for a better evaluation of the glycemic state of the patient.

A second improvement regards the quality of the food logs both in term of exhaustivity, by

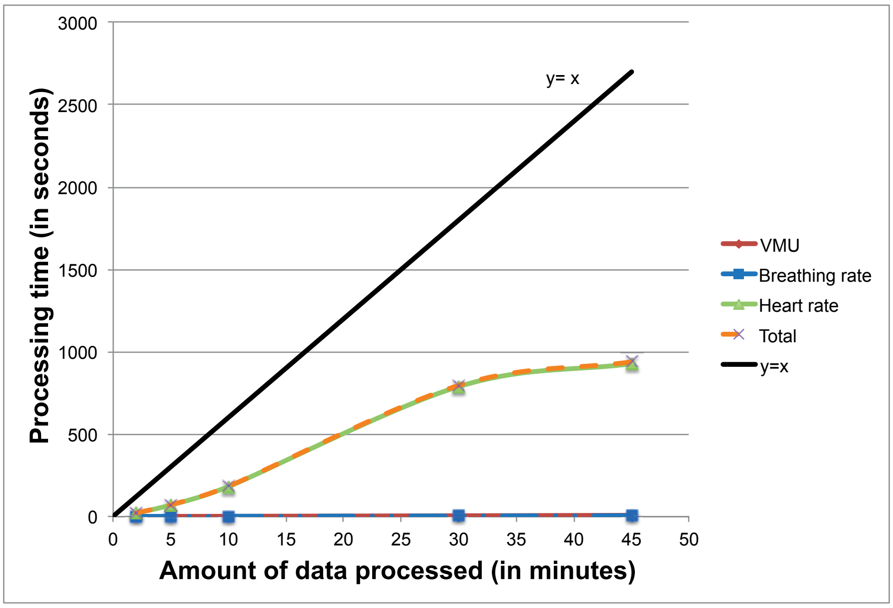


Figure 7.9 – Processing time to determine energy expenditure

having the complete list of food ingested by the patient, and in term of accuracy, by evaluating more precisely the composition of the food. Relying on self-reporting from the patients about their food intake can lead to discrepancy in number of meals reported. Furthermore, the estimation of the nutritional value of the food is an approximation as this value depends on the way the food is prepared which cannot be captured by the food databases available. Improving the quality of the food logs is expected to improve the accuracy of our model.

Finally, the performance evaluation confirmed that our model is compatible with a mobile environment and that the estimation of the blood glucose level of a patient can be computed without having to share data with external services.

8 Conclusions

In this thesis we focused on the detection of complex events using wearable sensors. The amount of digital data that is generated by users and about the users is unprecedented and an important share of this data comes from wearable sensors which are continuously sensing the user's states and her environment. We evaluated how this information can be harnessed to automatically detect complex events in order to assist the user. We defined a general mobile framework to generate complex events based on wearable sensors. This framework starts by detecting simple events from noisy raw data. These events are subsequently augmented with a meaning, a semantic value. We capitalise on this value to generate more complex events capable of representing intricate states of the user.

We applied the framework for generating complex events in two different contexts. For each context we devised and implemented the different components of the framework and we evaluated our approach against real life datasets.

The first context was presented in part I of this thesis. We studied how to enable smartphone to build a representation of the user's life in a similar fashion than the human brain stores episodic memories. These memories are constructed in a hierarchical approach, going from the most events-near episodic elements, to more abstract views such as complex episodic memories. We applied this representation in the context of smartphones where episodic elements are assimilated to sensor readings. These elements are hierarchically aggregated until reaching the granularity of routines of the user which can be assimilated to the complex episodic memories.

The second context was presented in part II. We advanced the field of non-invasive glucose monitoring by studying the possibility of using mobile sensors coming from off-the-shelf devices to detect glycaemic events. These sensors acquire signals in uncontrolled settings as they are meant to be worn in everyday life by the user. These settings lead to signals that are much more noisy than the type of signals that are usually used to evaluate alternative non-invasive glucose monitoring approaches, these signals are usually acquired in an hospital where noise such as movement noise are limited. We presented a mobile processing pipeline to estimate the glucose level of a patient based on physiological features. This pipeline manages to achieve comparable results to the state of the art approaches despite the original signal quality and the limited resources available to perform computation. We also proposed an approach based on an energy model which evaluates the blood glucose level of a patient based on her various sources of glucose and her glucose elimination.

For both contexts, we described how we apply a hierarchical framework for complex events recognition using wearable sensors. Although the framework provides generic steps that can be applied across contexts, the specificity of each domain necessitates the ad-hoc selection of the algorithms that will be used at each step. We summarise below each steps of the framework presented in Figure 1.2, for which the flow of data follows a bottom up approach.

Mobile signal processing Both D1namo and MemorySense employ filtering techniques to handle the noise of the raw data acquired by the sensors. In this thesis, we showed that multiple signal processing tools can be applied in a mobile environment. From standard passband filters, to more advanced adaptive filters, as well as clustering techniques which can be seen as filters retaining only a high frequency view of the underlying signal. This component alleviates the noise inherent to wearable sensors.

Simple event recognition This first layer of event recognition allows to add semantics to the data by detecting simple events. In MemorySense, we proposed the generation of SEMs using a modified version of the ESOINN algorithm to detect PoIs. Its performance were shown on par with the state of the art, while its incremental approach ensured that its runtime would remain low, even on the long run. In order to relieve the user from providing the semantics of each PoIs, we leverage an external data source in order to obtain the semantic of the place. In D1namo, the detection of QRS complexes can be seen as simple events in the physiological approach, while in the energy model, these simple events are represented as the energy expenditure, and more specifically the detection of period of low and high energy expenditure. While the QRS complexes are extracted using the approach proposed by Yazdani ([124]), the energy expenditure is detected for 3 separate underlying signals: heart rate, breathing rate and VMU. Given the limited semantic space used by both type of events, no external data source is required to label these events.

Semantic noise handling The semantic noise corresponds to simple events that are harmful for the detection of more complex events. In MemorySense this noise corresponds to SEMs which confuses the detection of CEMs. We proposed two approaches to handle such noise. The first one, based on finite state machines ensures that up to θ-noisy SEMs can be presented without affecting the detection of CEM, the second one, bases on frequent pattern mining, prefers to handle variations (considered as noise) in the sequences of SEMs by focusing on the detection of frequent patterns, effectively discarding noisy and infrequent SEMs. In D1namo, the semantic noise is handled in the physiological model once the heartbeats have been classified as belonging to hypoglycaemia or not. A sliding window over this classification ensures that noise in the classification is filtered out. This approach is based on the fact that glycemic events have a momentum and can (literally) not change in a heartbeat. In the energy model, the energy expenditures previously computed for the 3 underlying signals are combined by averaging only the closest pair of energy expenditure signals. This way, the noise introduced by external factors influencing differently the 3 underlying signals is alleviated.

Complex event recognition Finally, the complex events recognition is performed in MemorySense by the actual classification of sequences of SEMs into CEMs, our approach is evaluated and yields good results regarding the accuracy of the classification. In D1namo, the complex events are the glycaemic events themselves. The physiological model evaluation shows that it performs as good as the state of the art. The energy model is based on the balancing of the energy intake of the user from her meals, her energy expenditure and her insulin intake. The model is evaluated in term of fitness compared to a continuous glucose monitor. The results show that the energy model can represent the general trend of the glucose level of the patient (and therefore her glycaemic events), but some variations cannot be captured due to the simplicity of the model. Furthermore, the model relies on the proper manual log of the food by the patient.

Privacy preservation and energy efficient computation By design, the approaches taken in MemorySense and D1namo aim to simply prevent the sensors' data to leave the device. For this reason, efficient algorithms and processes are required to be able to perform complex computation on device. A key characteristic for such algorithms is to be incremental. In this way, the processing of new generated data is not be impacted (or with minimal impact) by the amount of data that has already been processed. Through this thesis, we built the different systems in an incremental fashion. The evaluation of the runtime performances on mobile of the different components of our systems validates the rationality of our approach.

8.1 Outlooks

The experience that we gained during this thesis allowed us to identify interesting research directions to pursue in the future.

Mobile neural networks Mobile neural networks recently opened the door to a wide range of powerful algorithms that can be used to detect complex events. Mobile GPUs on recent smartphones allow moderate parallel computation, and already enabled the use of small neural networks (see for example [89, 112]). Such approaches have the advantage of handling the fusion of signals more easily. However, while pre-trained networks can be used directly by mobile devices, it is still a challenge to train the network in a computationally restricted environment. Increasing the trainability of neural networks on mobile device would open the door to new possibilities including more powerful complex event detection. A second challenge of neural networks concerns their reduced interpretability. These networks are often assimilated to black boxes and the reason of their performance is still unclear. This can be problematic especially in the medical domain where diagnostics are built on interpretable observations. Developing an understanding of the models and not only their outcome would be valuable and justify the use of such algorithms for the medical domain.

Model sharing Another aspect that is left as future work is the transfer of the model learned between user. In this thesis, we took the decision of performing strict on-device computation. While this approach allows the development of models tailored for the user, it requires a bootstrapping phase. Instead, the transfer of models between users could be beneficial as it would allow to train the algorithm on larger datasets. In the case of MemorySense for instance, sharing routines between users, would allow to discover these routines faster for new users and could reduce the user's manual input of the activity she was performing by providing information coming from other users. However, this transfer raises questions regarding the processing of the user's data in a privacy preserving fashion to contribute to a more global model. Privacy schemes such as differential privacy can be applied in a distributed context to build a common dataset in which none of the users could be uniquely identified. The approaches taken by De Montjoye with openPDS [37] and Erlingsson [42] are of particular interest for this task. Studying the utility of such systems with regards to the detection of complex events can be beneficial for users, but also in general, for a better understanding of the events at hand.

The specific work we performed on MemorySense and D1namo also motivated research questions in each area that are left as future work.

MemorySense The vision of the episodic memory system by Conway defines EEs, SEMs, CEMs and their respective frames as structuring elements for the system. It would be interesting to study how mobile systems could use other function of the human brain, such as the semantic memory system and its property to build efficient information systems. Another aspect mentioned by Conway is the affective dimension of episodic memories. Collecting emotions about memories through mobile sensors such as camera while taking a self-portrait or microphone during a phone call, is a interesting research direction to enable new ways of retrieving memories.

D1namo In the evaluation of the energy model proposed in Chapter 8, we described that the accuracy of the model was dependant on the patients keeping a thorough log of their food intake. Improving the quality of the logs would also improve the quality of our model. This could be done either by defining incentives for the user to log all her food intake, or by automating this log. The energy model that we presented only covers the 3 dominant sources and sinks of glucose in the blood. It would be interesting to study how other aspect of the glucose dynamics such as its endogeneous production could be detected non-invasively and integrated in the energy model.

Bibliography

[1] K. Aberer, M. Catasta, H. Radu, J.-E. Ranvier, M. Vasirani, and Z. Yan. Memorysense: Reconstructing and ranking user memories on mobile devices. In *PERCOM Workshops '14*. IEEE, 2014.

[2] V. Agarwal, N. Banerjee, D. Chakraborty, and S. Mittal. Usense–a smartphone middleware for community sensing. In *MDM*, 2013.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[4] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[5] N. Aharony, A. Gardner, C. Sumter, and A. Pentland. Funf: Open sensing framework, 2011.

[6] B. E. Ainsworth, W. L. Haskell, S. D. Herrmann, N. Meckes, D. R. Bassett Jr, C. Tudor-Locke, J. L. Greer, J. Vezina, M. C. Whitt-Glover, and A. S. Leon. 2011 compendium of physical activities: a second update of codes and met values. *Medicine and science in sports and exercise*, 43(8):1575–1581, 2011.

[7] M. Altini, J. Penders, R. Vullers, and O. Amft. Estimating energy expenditure using body-worn accelerometers: a comparison of methods, sensors number and positioning. *IEEE journal of biomedical and health informatics*, 19(1):219–226, 2015.

[8] Amazon. Mechanical Turk. https://www.mturk.com, 2005. Accessed: 2017-01-01.

[9] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod record*, volume 28, pages 49–60. ACM, 1999.

[10] Apple. HealthKit apple developer. http://developer.apple.com/healthkit/, 2014. Accessed: 2017-01-01.

[11] D. Arvidsson, F. Slinde, and L. Hulthén. Free-living energy expenditure in children using multi-sensor activity monitors. *Clinical nutrition*, 28(3):305–312, 2009.

[12] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous computing*, 7(5):275–286, 2003.

[13] A. D. Association et al. Physical activity/exercise and diabetes. *Diabetes care*, 27(suppl 1):s58–s62, 2004.

[14] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom*, 2009.

[15] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context-aware places of interest recommendations for mobile users. In *International Conference of Design, User Experience, and Usability*, pages 531–540. Springer, 2011.

[16] N. Bicocchi, G. Castelli, M. Mamei, A. Rosi, and F. Zambonelli. Supporting location-aware services for mobile users with the whereabouts diary. In *MOBILWARE*, page 6, 2008.

[17] U. Blanke and B. Schiele. Daily routine recognition through activity spotting. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, pages 192–206. Springer-Verlag, 2009.

[18] S. Brage, N. Brage, P. Franks, U. Ekelund, and N. Wareham. Reliability and validity of the combined heart rate and movement sensor actiheart. *European journal of clinical nutrition*, 59(4):561–570, 2005.

[19] M. D. Breton et al. Adding heart rate signal to a control-to-range artificial pancreas system improves the protection against hypoglycemia during exercise in type 1 diabetes. *Diabetes technology & therapeutics*, 16(8):506–511, 2014.

[20] V. Bush et al. As we may think. *The atlantic monthly*, pages 101–108, 1945.

[21] K. Y. Chan, S.-H. Ling, T. S. Dillon, and H. T. Nguyen. Diagnosis of hypoglycemic episodes using a neural network based rule discovery system. *Expert Systems with Applications*, 38(8):9799–9808, 2011.

[22] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Nips*, volume 31, pages 1–9, 2009.

[23] D. Chen and H. Zhao. Data security and privacy protection issues in cloud computing. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 1, pages 647–651. IEEE, 2012.

[24] J. Chen, H. Guo, W. Wu, and W. Wang. imecho: an associative memory based desktop search system. In *CIKM*, pages 731–740. ACM, 2009.

[25] D. W. Cheung, J. Han, V. T. Ng, and C. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, pages 106–114. IEEE, 1996.

[26] B. Chikhaoui, S. Wang, and H. Pigot. Adr-splda: Activity discovery and recognition by combining sequential patterns and latent dirichlet allocation. *Pervasive and Mobile Computing*, 2012.

[27] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. A. Landay, J. Lester, D. Wyatt, and D. Haehnel. The mobile sensing platform: An embedded activity recognition system. *PerCom '08*, 2008.

[28] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE transactions on software engineering*, (3):178–187, 1978.

[29] C. C. Christensen, H. Frey, E. Foenstelien, E. Aadland, and H. E. Refsum. A critical evaluation of energy expenditure estimates based on individual o2 consumption/heart rate curves and average daily heart rate. *The American journal of clinical nutrition*, 37(3):468–472, 1983.

[30] S. Cichosz. A novel algorithm for prediction and detection of hypoglycemia based on continuous glucose monitoring and heart rate variability in patients with type 1 diabetes. *J. diabetes science and technology*, 2014.

[31] S. Cichosz et al. Combining information of autonomic modulation and cgm measurements enables prediction and improves detection of spontaneous hypoglycemic events. *J. Diabetes science and technology*, 2014.

[32] M. A. Conway. Episodic memories. *Neuropsychologia*, 2009.

[33] B. Cvetković, R. Milić, and M. Luštrek. Estimating energy expenditure with multiple models using different wearable sensors. *IEEE journal of biomedical and health informatics*, 20(4):1081–1087, 2016.

[34] M. Czerwinski and E. Horvitz. An investigation of memory for daily computing events. In *People and Computers XVI-Memorable Yet Invisible*, pages 229–245. Springer, 2002.

[35] C. Dalla Man, M. D. Breton, and C. Cobelli. Physical activity into the meal glucose—insulin model of type 1 diabetes: In silico studies. *Journal of diabetes science and technology*, 3(1):56–67, 2009.

[36] C. Dalla Man, R. A. Rizza, and C. Cobelli. Mixed meal simulation model of glucose-insulin system. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 307–310. IEEE, 2006.

[37] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland. openpds: Protecting the privacy of metadata through safeanswers. *PloS one*, 9(7):e98790, 2014.

[38] K. M. Diaz, D. J. Krupka, M. J. Chang, J. Peacock, Y. Ma, J. Goldsmith, J. E. Schwartz, and K. W. Davidson. Fitbit®: An accurate and reliable device for wireless physical activity tracking. *International journal of cardiology*, 185:138–140, 2015.

[39] S. Ding and M. Schumacher. Sensor monitoring of physical activity to improve glucose management in diabetic patients: A review. *Sensors*, 16(4):589, 2016.

[40] N. Eagle and A. S. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[41] K. Ellis, J. Kerr, S. Godbole, and G. Lanckriet. Multi-sensor physical activity recognition in free-living. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 431–440. ACM, 2014.

[42] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[44] K. Farrahi and D. Gatica-Perez. Discovering human routines from cell phone data with topic models. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pages 29–32. IEEE, 2008.

[45] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Trans. Intell. Syst. Technol.*, 2(1):3:1–3:27, Jan. 2011.

[46] D. Feldman, A. Sugaya, C. Sung, and D. Rus. idiary: From gps signals to a text-searchable diary. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 6. ACM, 2013.

[47] C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

[48] Fitbit. Fitbit activity tracker. http://www.fitbit.com/one, 2012. Accessed: 2017-01-01.

[49] Foursquare. Foursquare. http://foursquare.com, 2009. Accessed: 2017-01-01.

[50] S. Furao, T. Ogura, and O. Hasegawa. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*, 20, 2007.

[51] E. A. Gale. The rise of childhood type 1 diabetes in the 20th century. *Diabetes*, 51(12):3353–3361, 2002.

[52] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: fulfilling the memex vision. In *ACM Multimedia*, 2002.

[53] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *SIGKDD '07*, KDD '07, pages 330–339, New York, NY, USA, 2007. ACM.

[54] Google. Google Translate. http://translate.google.com, 2006. Accessed: 2017-01-01.

[55] Google. Google navigation system. https://play.google.com/store/apps/details?id=com.google.android.apps.maps, 2010. Accessed: 2017-01-01.

[56] Google. Google Places. https://developers.google.com/places/, 2010. Accessed: 2017-01-01.

[57] B. Guo, Z. Yu, X. Zhou, and D. Zhang. Memphone: From personal memory aid to community memory sharing using mobile tagging. In *Work in Progress session at PerCom '13*, 2013.

[58] B. Guo, D. Zhang, D. Yang, Z. Yu, and X. Zhou. Enhancing memory recall via an intelligent social contact management system. *Human-Machine Systems, IEEE Transactions on*, 44(1):78–91, 2014.

[59] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[60] B. H. Hansen, E. Kolle, S. M. Dyrstad, I. Holme, and S. A. Anderssen. Accelerometer-determined physical activity in adults and older people. *Medicine and science in sports and exercise*, 44(2):266–272, 2012.

[61] Y. He and Y. Li. Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[62] E. B. Hekler, M. P. Buman, L. Grieco, M. Rosenberger, S. J. Winter, W. Haskell, and A. C. King. Validation of physical activity tracking via android smartphones compared to actigraph accelerometer: laboratory-based and free-living validation studies. *JMIR mHealth and uHealth*, 3(2):e36, 2015.

[63] P. Herrero, P. Georgiou, N. Oliver, D. G. Johnston, and C. Toumazou. A bio-inspired glucose controller based on pancreatic β-cell physiology. *Journal of diabetes science and technology*, 6(3):606–616, 2012.

[64] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. Sensecam: A retrospective memory aid. In *UbiComp 2006: Ubiquitous Computing*, pages 177–193. Springer, 2006.

[65] P. Hong, M. Turk, and T. S. Huang. Gesture modeling and recognition using finite state machines. In *Automatic face and gesture recognition, 2000. proceedings. fourth ieee international conference on*, pages 410–415. IEEE, 2000.

[66] R. Humphrey. Clinical applications: The exercise caloric challenge. *ACSM's Health & Fitness Journal*, 10(2):40–41, 2006.

[67] J. Joseph, M. Torjman, J. Reich, A. Furnary, S. Nasraway, M. McNamara-Cullinane, D. Olson, and D. Walton. Evaluation of symphony cgm, a non-invasive, transdermal

continuous glucose monitoring system for use in critically ill patients. *Critical Care*, 18(1):1, 2014.

[68] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):58–68, 2005.

[69] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *ICPS*, 2010.

[70] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[71] T. Laitinen et al. Electrocardiographic alterations during hyperinsulinemic hypoglycemia in healthy subjects. *Annals of Noninvasive Electrocardiology*, 13(2):97–105, 2008.

[72] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Proceedings of the Workshop on the Nokia Mobile Data Challenge*, 2012.

[73] M. B. Lerner, N. Kybert, R. Mendoza, R. Villechenon, M. A. B. Lopez, and A. C. Johnson. Scalable, non-invasive glucose sensor based on boronic acid functionalized carbon nanotube transistors. *Applied Physics Letters*, 102(18):183113, 2013.

[74] P. Li, L. Yu, Q. Fang, and S.-Y. Lee. A simplification of cobelli's glucose–insulin model for type 1 diabetes mellitus and its fpga implementation. *Medical & biological engineering & computing*, pages 1–15, 2015.

[75] S. Ling, H. T. Nguyen, et al. Hypoglycaemia detection for type 1 diabetic patients based on ecg parameters using fuzzy support vector machine. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2010.

[76] S. H. Ling, P. P. San, H. K. Lam, and H. T. Nguyen. Non-invasive detection of hypoglycemic episodes in type 1 diabetes using intelligent hybrid rough neural system. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1238–1242. IEEE, 2014.

[77] S. H. Ling, P. P. San, and H. T. Nguyen. Non-invasive hypoglycemia monitoring system using extreme learning machine for type 1 diabetes. *ISA transactions*, 64:440–446, 2016.

[78] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Sound-sense: Scalable sound sensing for people-centric applications on mobile phones. In *MobiSys*, 2009.

[79] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, 2010.

[80] M. S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments, & Computers*, 32(1):93–110, Mar. 2000.

[81] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell. Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in sports and exercise*, 45(11):2193, 2013.

[82] P. Maragos and R. W. Schafer. Morphological systems for multidimensional signal processing. *Proceedings of the IEEE*, 78(4):690–710, 1990.

[83] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[84] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[85] R. Montoliu and D. Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[86] T. Nakata. Recognizing human activities in video by multi-resolutional optical flows. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1793–1798. IEEE, 2006.

[87] T. Nguyen, D. Phung, S. Gupta, and S. Venkatesh. Extraction of latent patterns and contexts from social honest signals using hierarchical dirichlet processes. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 47–55, Mar. 2013.

[88] J. F. Nichols, C. G. Morgan, J. A. Sarkin, J. F. Sallis, and K. J. Calfas. Validity, reliability, and calibration of the tritrac accelerometer as a measure of physical activity. *Medicine and science in sports and exercise*, 31(6):908–912, 1999.

[89] F. J. Ordóñez and D. Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[90] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 299–314. ACM, 2010.

[91] D. Peebles, H. Lu, N. D. Lane, T. Choudhury, and A. T. Campbell. *Community-Guided Learning: Exploiting Mobile Sensor Users to Model Human Behavior.* 2010.

[92] S. M. Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[93] K. Plis, R. Bunescu, C. Marling, J. Shubrook, and F. Schwartz. A machine learning approach to predicting blood glucose levels for diabetes management. *Modern Artificial Intelligence for Health Analytics. Papers from the AAAI-14*, 2014.

[94] J. Ranvier, M. Catasta, I. Gavrilovic, G. Christodoulou, H. Radu, T. Signo', and K. Aberer. Memoit: From lifelogging application to research platform. In S. Mukherjea, editor, *Mobile Application Development, Usability, and Security*, chapter 1, pages 1–24. IGI global, Hershey, PA, 2017.

[95] J.-E. Ranvier, M. Catasta, M. Vasirani, and K. Aberer. Routinesense: A mobile sensing framework for the reconstruction of user routines. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 150–159. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015.

[96] J.-E. Ranvier, F. Dubosson, J.-P. Calbimonte, and K. Aberer. Detection of hypoglycemic events through wearable sensors. In *Proceedings of the International Workshop on Semantic Web Technologies for Mobile and PErvasive Environments 2016*, number EPFL-CONF-218544. CEUR-WS, 2016.

[97] N. Ravi et al. Activity recognition from accelerometer data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[98] E. Ravussin, S. Lillioja, T. E. Anderson, L. Christin, and C. Bogardus. Determinants of 24-hour energy expenditure in man. methods and results using a respiratory chamber. *Journal of Clinical Investigation*, 78(6):1568, 1986.

[99] R. Rawassizadeh, M. Tomitsch, K. Wac, and A. M. Tjoa. Ubiqlog: a generic mobile phone-based life-log framework. *Pers Ubiquit Comput*, 17(4):621–637, 2013.

[100] K. Rebrin, N. F. Sheppard Jr, and G. M. Steil. Use of subcutaneous interstitial fluid glucose to estimate blood glucose: revisiting delay and sensor offset, 2010.

[101] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, 2010.

[102] rjmarsan. Mobile Weka. https://github.com/rjmarsan/Weka-for-Android, 2010. Accessed: 2017-01-01.

[103] N. A. B. A. Salam, W. H. bin Mohd Saad, Z. B. Manap, and F. Salehuddin. The evolution of non-invasive blood glucose monitoring system for personal application. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(1):59–65, 2016.

[104] P. P. San, S. H. Ling, H. Nguyen, et al. Evolvable rough-block-based neural network and its biomedical application to hypoglycemia detection system. *IEEE transactions on cybernetics*, 44(8):1338–1349, 2014.

[105] P. P. San, S. H. Ling, and H. T. Nguyen. Deep learning framework for detection of hypoglycemic episodes in children with type 1 diabetes. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 3503–3506. IEEE, 2016.

[106] D. A. Santos, A. M. Silva, C. N. Matias, J. P. Magalhães, D. A. Fields, C. S. Minderico, U. Ekelund, and L. B. Sardinha. Validity of a combined heart rate and motion sensor for the measurement of free-living energy expenditure in very active individuals. *Journal of Science and Medicine in Sport*, 17(4):387–393, 2014.

[107] D. Schoeller and E. Van Santen. Measurement of energy expenditure in humans by doubly labeled water method. *Journal of Applied Physiology*, 53(4):955–959, 1982.

[108] M. Senior. Novartis signs up for google smart lens. *Nature biotechnology*, 32(9):856–856, 2014.

[109] E. Smalley. Medtronic automated insulin delivery device gets fda nod, 2016.

[110] Sobel. Accuracy of a novel noninvasive multisensor technology to estimate glucose in diabetic subjects during dynamic conditions. *J. of diabetes science and technology*, 2014.

[111] L. Soltanzadeh, A. Taheri, and M. Rabiee. Implementing a prototype diabetic telemedicine system. *International Journal of Computer Science Issues (IJCSI)*, 11(4):86, 2014.

[112] I. Song, H.-J. Kim, and P. B. Jeon. Deep learning for real-time robust facial expression recognition on a smartphone. In *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, pages 564–567. IEEE, 2014.

[113] R. Srikant and R. Agrawal. *Mining sequential patterns: Generalizations and performance improvements*. Springer, 1996.

[114] M. Stenerson, F. Cameron, S. R. Payne, S. L. Payne, T. T. Ly, D. M. Wilson, and B. A. Buckingham. The impact of accelerometer use in exercise-associated hypoglycemia prevention in type 1 diabetes. *Journal of diabetes science and technology*, 9(1):80–85, 2014.

[115] M. Stenerson, F. Cameron, D. M. Wilson, B. Harris, S. Payne, B. W. Bequette, and B. A. Buckingham. The impact of accelerometer and heart rate data on hypoglycemia mitigation in type 1 diabetes. *Journal of diabetes science and technology*, 8(1):64–69, 2014.

[116] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2), 2008.

[117] S. Stipkovic, R. Bruns, and J. Dunkel. Pervasive computing by mobile complex event processing. In *e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on*, pages 318–323. IEEE, 2013.

[118] K. Turksoy et al. Multivariable adaptive closed-loop control of an artificial pancreas without meal and activity announcement. *Diabetes technology & therapeutics*, 15(5):386–400, 2013.

[119] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *MobiSys*, 2009.

[120] Z. Xiao and Y. Xiao. Security and privacy in cloud computing. *IEEE Communications Surveys & Tutorials*, 15(2):843–859, 2013.

[121] Y. Xu, N. Stojanovic, L. Stojanovic, and D. Kostic. An approach for dynamic personal monitoring based on mobile complex event processing. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, page 464. ACM, 2013.

[122] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *EDBT*, 2011.

[123] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *ISWC '12*, pages 17–24. Ieee, 2012.

[124] S. Yazdani and J.-M. Vesin. Adaptive mathematical morphology for qrs fiducial points detection in the ecg. In *Computing in Cardiology Conference*, pages 725–728, 2014.

[125] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 1–10. IEEE, 2009.

[126] M. Zanon, G. Sparacino, A. Facchinetti, M. Riz, M. S. Talary, R. E. Suri, A. Caduff, and C. Cobelli. Non-invasive continuous glucose monitoring: improved accuracy of point and trend estimates of the multisensor system. *Medical & biological engineering & computing*, 50(10):1047–1057, 2012.

[127] M. Zanon, G. Sparacino, A. Facchinetti, M. S. Talary, M. Mueller, A. Caduff, and C. Cobelli. Non-invasive continuous glucose monitoring with multi-sensor systems: A monte carlo-based methodology for assessing calibration robustness. *Sensors*, 13(6):7279–7295, 2013.

[128] Zephyr. Zephy bioharness. http://www.zephyranywhere.com/products/bioharness-3, 2016. Accessed: 2017-01-01.

[129] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.

[130] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *WWW '08*, pages 247–256. ACM, 2008.

[131] Y. Zheng, L. Wang, R. Zhang, X. Xie, and W.-Y. Ma. Geolife: Managing and understanding your past life over maps. In *Proceedings of the The Ninth International Conference on Mobile Data Management*. IEEE Computer Society, 2008.

JEAN-EUDES RANVIER

Born in France, 21 September 1988

jeranvier@gmail.com

Route de Cossonay, 19
1008 - Prilly (Switzerland)

+41 (0) 789 21 83 08

EDUCATION

2012-2017 Ecole polytechnique fédérale de Lausanne, Switzerland

PhD I did my PhD on *Complex Event Recognition through Wearable Sensors* in the laboratory of distributed information systems of Karl Aberer.
courses outline: Advanced databases, Information theory and coding, Games theory for crowd and networks

2008–2012 Université de Technologie de Troyes, France

Engineering degree Master student in the department of information systems of UTT with a minor in software project management and distributed information systems.
courses outline: Services oriented architecture, Software quality, Management of computer-based projects, Signal analysis and processing I, Various environmental courses

Fall 2009 University of Oulu, Finland

Exchange semester Exchange semester in the department of electrical engineering of the university of Oulu.
courses outline: Software engineering, multimedia systems, Digital image processing

WORK EXPERIENCE

2012-2016 Teaching assistant at EPFL, Switzerland

Teaching Leading exercises sessions of various courses: Introduction to object oriented programming, Network security, Distributed information systems, Physics 101

2011-2012 Intern at EPFL, Switzerland

Internship The research topic was to build a decentralized recommender system based on a combination of content-based recommendations and collaborative filtering. The system is assessing the credibility of webpages for a specific user based on features of these pages and ratings provided by her social network.

Spring 2010 IT Manager at Lisa Leleu Studios, USA

Internship Web development and network management for a design company located in Doylestown, PA.

2008 - 2010 Youth leader at Telligo, France

Summer Job Organisation of scientific workshops and games for adolescents and management of daily life activities during multiple summer camps and educational schooltrips.

SKILLS

Operating systems	OSX, UNIX/LINUX, Windows
Programming	Java, Android, C, HTML, CSS, PHP, javascript/coffeeScript, XML, Bash, Python
Concepts	OOP, Web services, P2P, Distributed data processing, Recommender systems, Data mining
Tools	Weka, Eclipse, Android studio, Node.js Git, SVN, Terminal, Latex, Matlab, Photoshop & Illustrator

PUBLICATIONS

2017 Jean-Paul Calbimonte et al.
Semantic Representation and Processing of Hypoglycemic Events Derived from Wearable Sensor Data
Journal of Ambient Intelligence and Smart Environments

Jean-Eudes Ranvier et al.
MEmoIt: From Lifelogging Application to Research Platform
Book chapter in "Mobile Application Development, Usability, and Security"

2016 Fabien Dubosson et al.
D1NAMO, A Personal Health System for Glycemic Events Detection
Artificial Intelligence for Diabetes

Jean-Eudes Ranvier et al.
Detection of Hypoglycemic Events through Wearable Sensors
Workshop on Semantic Web Technologies for Mobile and Pervasive Environments

2015 Jean-Eudes Ranvier et al. - *Best paper award*
RoutineSense: A Mobile Sensing Framework for the Reconstruction of User Routines.
Mobiquitous

2014 Michele Catasta, et al.
B-hist: Entity-centric search over personal web browsing history
Journal of Web Semantics

Karl Aberer et al.
Memo-it: don't write your diary, sense it.
UbiComp Adjunct

Karl Aberer et al.
MemorySense: Reconstructing and ranking user memories on mobile devices.
PerCom Workshops

2012 Thanasis G. Papaioannou et al.
A decentralized Recommender System for Effective Web Credibility Assessment.
CIKM

OTHER INFORMATION

Languages French · Mother tongue
English · Fluent
German · Beginner

Interests Saxophone · Badminton · Chess · Coding pet projects from raytracer engine to library of useful tools