A COMPUTATIONAL FRAMEWORK FOR HUMAN-CENTERED MULTIMODAL DATA ANALYSIS

by

VANGELIS METSIS

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2011

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisors Dr. Fillia Makedon and Dr. Heng Huang for the invaluable guidance and support they have provided to me during the course of my PhD study.

I would also like to deeply thank my committee members, Dr. Vassilis Athitsos and Dr. Bahram Khalili. My discussions with them had a significant impact on my work.

Special thanks go to all the members of Heracleia Lab for making my days interesting and challenging at the same time. Their presence made the working environment more friendly and hospitable.

Last but not least, I would like to thank my parents Theofanis and Athina, and my brother Laertis for always being there for me and making me feel that there were always on my side even though I was so far way from them.

November 22, 2011

ABSTRACT

A COMPUTATIONAL FRAMEWORK FOR HUMAN-CENTERED MULTIMODAL DATA ANALYSIS

VANGELIS METSIS, Ph.D.

The University of Texas at Arlington, 2011

Supervising Professors: Heng Huang and Fillia Makedon

Human-Centered computing defines a field of study in which computational processes affect the human being, either through ubiquitous and pervasive use of devices or any effect that improves the human condition. Human-Centered Computing applications face serious challenges in the handling of data collection, modeling, and analysis. Traditionally, the analysis of different aspects of human well-being derives from a variety of non-interrelated methods which has made it difficult to correlate and compare the different experimental findings for an accurate assessment of the contributing factors.

This dissertation describes new algorithms that enable more accurate and efficient multimodal data analysis of Human-Centered computing applications in order to improve decision-making in healthcare. In particular, this work provides a theoretical framework for multimodal and inter-related data analysis and demonstrates the theory in different cases where the purpose is to (a) monitor the health condition of the human subject, and (b) to improve the quality of life through the understanding of a subject's behaviors.

Our computational framework can efficiently analyze and interpret data of different modalities coming from the same human subjects. Emphasis is put on the evaluation of feature selection and classification techniques and their use for heterogeneous data fusion in order to improve the accuracy of the obtained results. Our experimental results show that the same basic methods can be used to analyze data regarding both the physiological and behavioral properties of a human subject, and to correlate the different findings into more meaningful and reliable information.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

LIST OF ILLUSTRATIONS ix

LIST OF TABLES xi

Chapter Page

1. INTRODUCTION 1

1.1 Problem 1

1.2 Motivation 2

1.3 Proposed work 4

2. COLLECTION OF HUMAN-CENTERED MULTIMODAL DATA 6

2.1 Introduction 6

2.2 Collection of Human Behavioral Data 7

2.2.1 Collection of Data to Monitor Sleep Patterns 8

2.2.2 Collection of Data to Monitor Medication Intake 11

2.2.3 Large-Scale Sensor Data Collection 12

2.2.4 Higher Level Data Collection 19

2.2.5 Ensuring the Quality of the Collected Data 22

2.3 Collection of Human Genomic and Physiological Data 24

2.3.1 HRMAS ^{1}H MRS Data 26

2.3.2 Gene Expression Data 28

2.3.3 Array Comparative Gene Hybridization (aCGH) Data 28

3. EFFICIENT FEATURE SELECTION IN HUMAN-CENTERED DATA 32

3.1 Introduction 32
3.2 Hybrid Sparsity Regularization (HSR) for Feature Selection in aCGH Data 33
3.3 Feature Selection Methodology 36
3.3.1 Hybrid Sparsity Regularization (HSR) 36
3.3.2 An Efficient Algorithm to Solve L2R21R2 39
3.3.3 Algorithm Analysis 40
3.3.4 Competitive Feature Selection Methods 43
3.4 Datasets 43
3.5 Experiments 44
3.5.1 Biomarker analysis 46
3.6 Discussion 50
4. ANALYSIS AND FUSION OF HETEROGENEOUS MULTIMODAL DATA 56
4.1 Introduction 56
4.2 Heterogeneous Data Fusion to Type Brain Tumor Biopsies 56
4.2.1 Problem 56
4.2.2 Datasets 57
4.2.3 Methods and Experimental Results 58
4.2.4 Biological Meaning 61
4.3 Non-Invasive Analysis of Sleep Patterns via Multimodal Sensor Input 62
4.3.1 Introduction 62
4.3.2 Related Work 64
4.3.3 Multimodal Sleep Pattern Analysis 65
4.3.4 Data Analysis and Classification 66
4.3.5 Discussion 74

5. DISCUSSION OF FRAMEWORK AND EVALUATION 76
5.1 Introduction 76
5.2 Evaluation of a Computational Framework for Assistive Environments 77
5.2.1 Functionality 77
5.2.2 Usability 80
5.2.3 Security and Privacy 82
5.2.4 Architecture 83
5.2.5 Intelligence 84
5.2.6 Quality of Service (QoS) 86
5.2.7 Cost 87
5.3 Discussion 88
6. CONCLUSION 90
REFERENCES 93
BIOGRAPHICAL STATEMENT 103

LIST OF ILLUSTRATIONS

Figure Page

1.1 Architecture of the proposed Computational Framework 4

2.1 A graphical representation of the simulated assisted living apartment at Heracleia Lab 8

2.2 An example of a subject lying on his side on the pressure mat (top) and the measurement values obtained (bottom) 10

2.3 A 3D representation of the input obtained by the Kinect depth sensor 11

2.4 The smart drawer system 13

2.5 Example of a factor graph 15

2.6 Example of messages being exchanged in the factor graph 16

2.7 Percentage of coverage versus number of agents when using a Gaussian map 17

2.8 Gaussian map coverage 18

2.9 Percentage of coverage versus percentage of agents failed, for a static and dynamic system. We can see that the dynamic system adapts well to failures 19

2.10 An example ontology centered middleware architecture 23

2.11 The proposed QoS negotiation mechanism 25

2.12 *Ex vivo* HRMAS ^{1}H MR spectrum of a 5.8 mg glioblastoma multiforme (GBM) tissue biopsy 27

2.13 An example of how Genes are expressed by being transcribed into RNA, and subsequently translated into proteins 29

2.14 The images in this figure visualize the CNVs of a sample of colorectal cancer with liver metastasis 31

3.1 Visualization of the coefficient table W after the application of HSR feature selection method on aCGH dataset 3 39

3.2 Classification accuracy results for datasets 1 and 2 comparing HSR (L2R21R2) with 6 existing feature selection methods using SVM and Logistic Regression classifiers 51

3.3 Classification accuracy results for datasets 3 and 4 comparing HSR (L2R21R2) with 6 existing feature selection methods using SVM and Logistic Regression classifiers 52

3.4 Genotype-Phenotype mapping of well known genes and diseases on Chromosome 17, extracted from Entrez Genome NCBI Database 53

3.5 Clone-Gene mapping in the region 33,080K-34,650K bp of Chromosome 17. In the genomic area covered by the examined clone (RP11-47L3) we find the gene SLFN5 53

4.1 Fusion feature selection and classification framework 59

4.2 Classification results with various combinations of data and feature selection methods 60

4.3 Detection of motion using the sum of absolute frame differences (S) and a threshold $T = 130$ 69

4.4 The 5 different body postures 71

5.1 The basic attributes of the framework 77

LIST OF TABLES

Table Page

3.1 The 20 most important BAC/PAC clones of Dataset 1 and the corresponding genes found in the genomic area covered by each clone. 48

3.2 The 20 most important BAC/PAC clones of Dataset 2 and the corresponding genes found in the genomic area covered by each clone. 49

3.3 The 20 most important BAC/PAC clones of Dataset 3 and the corresponding genes found in the genomic area covered by each clone. 54

3.4 The 20 most important BAC/PAC clones of Dataset 4 and the corresponding genes found in the genomic area covered by each clone. 55

4.1 Best results for each dataset and each classifier for the 6 class classification task. The feature selection method that achieved the highest accuracy along with the accuracy itself is shown in each table cell. 62

4.2 Classification accuracy results for Body Posture and Motion Type recognition. 75

5.1 Summary of attributes to evaluate a Human-Centered Computational Framework for Assistive Living. 89

6.1 Summary of problems solved by our Human-Centered Computational Framework and methods we proosed to solve them. 92

CHAPTER 1

INTRODUCTION

1.1 Problem

As the field of Computer Science advances, the focus of the researchers shifts from simply providing enhanced services to the humans, to improving their overall well being and quality of life by putting the humans themselves at the center of attention of the research and development process. In other words there is a trend towards what we call Human-Centered Computing [1]. Human-Centered computing defines a field of study in which computational processes affect the human being, either through ubiquitous and pervasive use of devices or any effect that improves the human condition. In this era of ubiquitous and mobile computing the aim of pervasive assistive technologies is to provide for independent living and improve the quality of life of people. The emphasis of ongoing research projects has been on providing the necessary services and integrating the following types of system goals:

1. the ability to recognize fast and accurately important changes to the environment, changing needs, events and patterns through on-site or remote monitoring using mobile and static sensors and software tools for automated data collection, fusion and analysis of heterogeneous environmental/health/behavioral data;
2. early event detection for the prevention of accidents, emergency response and decision support that helps make decisions as to the next step to take, alerts to generate or the actuation/activation of assistive devices; and
3. seamless access to home and external virtual and physical resources through an invisible and intelligent computing infrastructure that allows the human to

control and make changes to his/her physical/digital environment. The latter assumes the existence of easy to use communication interfaces with persons, objects and entities inside and outside the home.

Human-Centered Computing applications face serious data collection, modeling, analysis and synthesis challenges. One such challenge is to enable the efficient modeling and analysis of a plethora of multi-modal data collected from diverse human-based activities. Traditionally, the analysis of different aspects of human well-being has been based on a variety of non-interrelated methods which has made it difficult to correlate and compare the different experimental findings for an accurate assessment of the contributing factors. Tools are needed to make it possible to correlate, for example, the clinical state with the behavioral, the genotype with the phenotype or the psychosocial state with brain activation or neural condition as early and accurately as possible. This challenge is particularly important in pervasive environments rich in different types of sensors where the aim is at monitoring human activities implicitly.

1.2 Motivation

To date, there have been significant advancements in specific areas of Computer Science such as: Sensor Networks [2], Wireless Communications [3], Databases [4], Pattern Recognition and Machine Learning [5], Data Mining [6], Computer Vision [7], Robotics [8] and other areas which can facilitate the creation of a smart interactive environment adapted to assisted living. However, most previous works do not take into consideration the specific properties of the data originating from human behavior and physiology. Putting humans to the center of attention poses new challenges regarding security and privacy, intrusiveness and the special needs of groups with disabilities.

There is a need for a framework that will exploit the advancements in the different research areas by taking into consideration the special requirements deriving from direct interaction with humans and suggesting new advancements, where necessary, to meet these requirements. When monitoring human beings, it is very common to simultaneously obtain input from a variety of sensing devices. The amount of generated data is usually large and noisy. Therefore, methods to process the data, discard the noise and extract meaningful information are a necessity. Moreover, the different data modalities are not irrelevant to each other and in order to extract meaningful information from them there is a need for methods to combine or fuse these data or the features of interest extracted from them. Finally, the extracted information has to be of type that can be interpreted by the experts which in this case are physicians or doctors. That means the proposed computational methods have to be coupled with knowledge from Bioinformatics and Medical Informatics.

In assistive environments, the data collected come from two main sources: (a) the function of the monitored subject's body and (b) the activities they perform over time. The data collected from the first source are used to monitor the subject's health condition whereas the data collected from the second source are used to analyze behavioral patterns which in turn can either be related with certain health conditions or can just be used to facilitate everyday activities. The collected data can be analyzed either in real-time or off-line in order to extract useful information about the subjects being monitored. In addition, each of the above data major sources is further divided into a number of different data modalities each of which may require a different approach in order to become meaningful. These different modalities, although at first may seem unrelated, usually carry complementary information about their sources, which if strategically combined may become much more useful.

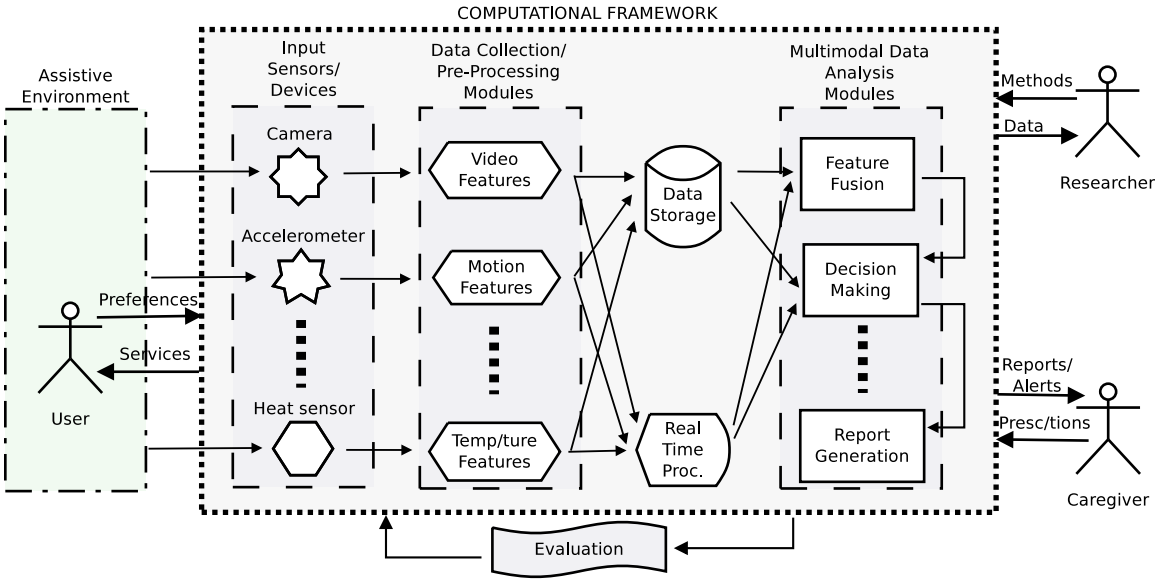


Figure 1.1. Architecture of the proposed Computational Framework for Human-Centered Multimodal Data Analysis.

1.3 Proposed work

In this work we present a computational framework to efficiently collect, analyze and interpret data of different modalities coming from the same human subjects. We use computational methods and algorithms originating from the field of Machine Learning and Pattern Recognition to discover useful patterns about human health condition and behavior as well as other methods that facilitate the use of Pervasive Technologies to the service of humans. The generated results include the biological meaning of our findings and can be easily interpreted by the physicians. We focus on the evaluation of feature selection and classification techniques and their use for heterogeneous data fusion in order to improve the accuracy of the obtained results and we show that the same basic methods can be used to analyze data regarding both the health condition of the monitored subject and their behavioral patterns. In addition, our methods take into consideration the specificalities of monitoring human

subjects and manage to be minimally invasive and privacy preserving. Finally, we examine the properties that will guarantee the success of such a framework in real life applications and we propose metrics to quantify the degree to which each of these properties achieves its goals. In summary, we suggest a framework that can unify the monitoring process and the analysis of data of different sources and modalities coming from human subjects. Figure 1.1 gives a general overview of the proposed Computational Framework for Human-Centered Multimodal Data Analysis.

The rest of this dissertation is organized as follows. Chapter 2 gives an overview of our work in human-centered data collection methodologies. In chapter 3 we present our work in feature selection from human-centric data. The methodology for fusion of different modalities of human-centric data is described in chapter 4. Chapter 5 presents our evaluation framework. Finally, chapter 6 summarizes and concludes this dissertation.

CHAPTER 2

COLLECTION OF HUMAN-CENTERED MULTIMODAL DATA

2.1 Introduction

Data input to information processing systems has been a problem as old as the problem of computing itself. In the past, data input types were limited to small number of different formats, such as text coming from a keyboard, or encoded data coming from card readers. With the introduction of multimedia devices, the range of input types started to expand and included audio, video and various other signals. More recently, the family of input devices has been extended by new members such as touch screens and various static or mobile sensors. Together with the the range of input types, there has been a big expand in the amount of data collected from different sources. This created a challenge not only in processing and storing the collected data, but also extracting meaningful information from them. Researchers have invested their efforts in creating better methods for both collecting data accurately and efficiently and analyzing the collected data.

The main focus of this dissertation is the development of methods to analyze multimodal data coming from human subjects, however, in this chapter, we also present a variety of methods that we have developed to collect the data that we have used in some of our experiments. Although the data modalities that we are interested in come from human subjects, the subjects themselves are not involved in the data input process. That means that the data need to be automatically collected and analyzed with minimal or no manual human effort. The collected data come

either from the natural interaction of the subjects with their environments, or from measurements regarding their physiological condition.

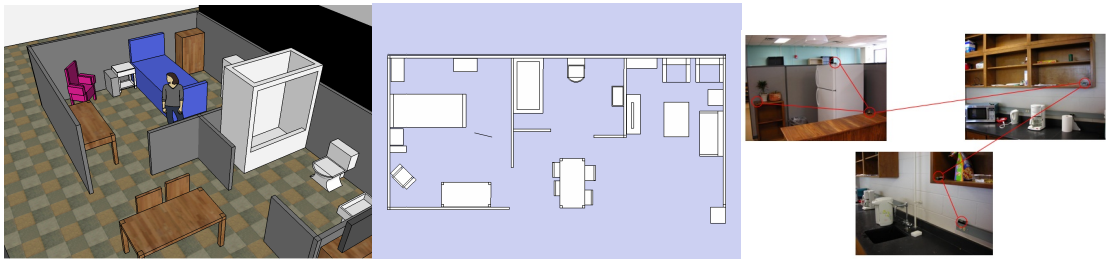
2.2 Collection of Human Behavioral Data

The collection of behavioral data from human subjects is a challenging task due to a number of reasons such as, the difficulty to anticipate and facilitate all the different states of the environment, the dynamic nature and conditions of the environments where the human beings live in and the sensing capabilities of the sensors/devices used to collect the data. The collected data can be limited to a specific task performed in a small predefined area, for example monitoring sleeping in bed, or can be more general and span a large percentage of the environment used by an average user during their daily routine activities, for example monitoring the exact location of a human inside an apartment.

At the Heracleia Human-Centered Laboratory[1], we have set up a simulated Assisted Living apartment. The apartment is covered by a variety of sensors that are used to monitor daily activities and provide assistive services to elderly or disabled people. Figure 2.1 shows a graphical representation of the apartment and an example of sensors used to detect the user's location at every moment.

In the remaining of this chapter, we will describe examples of sensor types and methodologies that we have developed to collect human behavioral data. In the next chapters, we will apply our methods to analyze some of these data and extract meaningful information from them. We will start by presenting methods tailored for data collection of some specific type of activity performed by the user and we extend to methods that handle more general (or high level) activities.

[1]For more information visit: http://heracleia.uta.edu/



(a) (b) (c)

Figure 2.1. A graphical representation of the simulated assisted living apartment at Heracleia Lab. (a) 3D representation. (b) 2D representation. (c) Examples of sensors placed in the apartment.

2.2.1 Collection of Data to Monitor Sleep Patterns[2]

The monitoring of sleep patterns is of major importance for various reasons such as, the detection and treatment of sleep disorders, the assessment of the effect of different medical conditions or medications on the sleep quality and the assessment of mortality risks associated with sleeping patterns in adults and children. Sleep monitoring by itself is a difficult problem due to both privacy and technical considerations. The proposed system uses a combination of non-invasive sensors to collect data about sleep patterns: a contact-based pressure mattress and a non-conntact 3D image acquisition device, which can can complement each other. To evaluate our system we used real data collected in Heracleia Lab's assistive living apartment.

For the needs of our experiments we collected data from 7 different individuals simulating their sleep habits. Each individual lied on the bed for a period of time and performed the actions that they would normally perform if they went to bed. That involved getting in bed, staying still for periods of time in different postures, changing body postures, moving parts of the body like the arms or the legs and getting out of the bed. The different actions performed during that period of time were recorded

[2]For more information about this project, the reader can refer to section 4.3

using 2 different sensors. The first one was a bed pressure mat (see section 2.2.1.1) that we put under the sheets, and the second one was a Microsoft Kinect sensor (see section 2.2.1.2) that we mounted on the ceiling. The recorded data were then manually annotated according to the various classes of interest, such body posture, motion occurrence, etc. In section 4.3 we will explain our methodology to analyze sleep patterns.

2.2.1.1 Data collected from FSA bed pressure mat

The FSA bed mat system produced by *Vista Medical Ltd* provides a $1920mm \times 762mm$ sensing area which contains an array of 32×32 pressure sensors. Each of the sensors can capture a measurement in the range 0 to 100 mmHg (1.93 PSI) with a scan frequency of up to 5 Hz. The measurements can be recorded over a period of time and can be exported as a set of time stamped vectors containing the values of each of the 1024 pressure sensors for each time stamp. To make visualization easier we can consider each of these vectors as a frame of a video. Each of the sensors can be considered as pixel of a gray-scale image with an intensity ranging from 1 to 100. Thus each frame can be considered as a 32 by 32 pixel image. Figure 2.2 illustrates a visualization example of the pressure values captured in one frame. The color coding is just a convention to facilitate visualization.

2.2.1.2 Data collected from Kinect

Kinect is a motion sensing input device designed by Microsoft for the Xbox 360 video game console [9]. Kinect outputs 3 different data streams, RGB video stream, depth sensing video stream and audio. The video output frame rate is 30 Hz. The RGB video stream uses 8-bit VGA resolution (640×480 pixels), while the monochrome depth sensing video stream is in VGA resolution (640×480 pixels) with

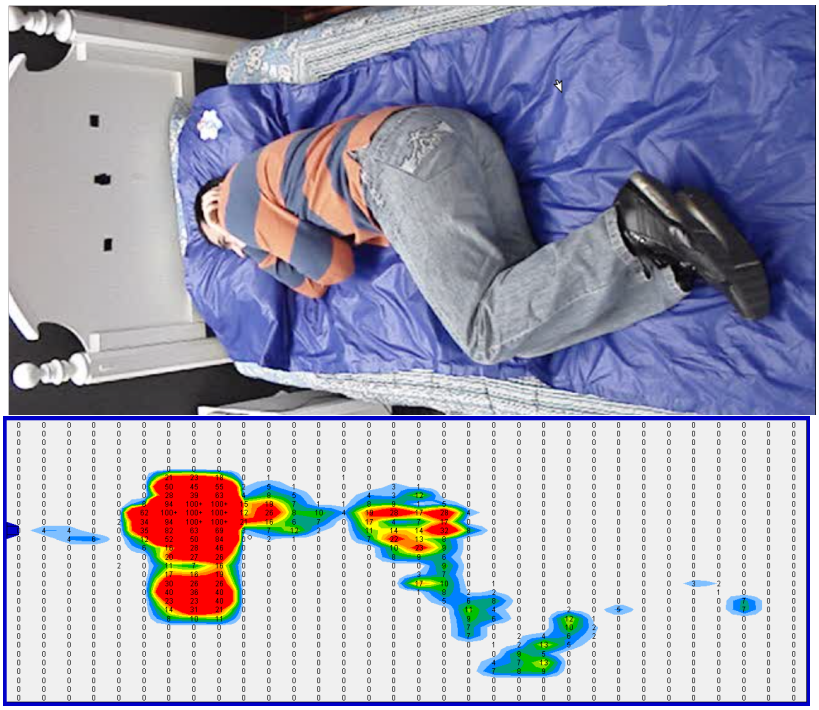


Figure 2.2. An example of a subject lying on his side on the pressure mat (top) and the measurement values obtained (bottom).

11-bit depth, which provides 2,048 levels of sensitivity. In our experiments we used only the depth sensing video stream. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. That feature makes the kinect usable even in very low lighting conditions, which is usually the case during the night sleep. Furthermore, the 3D input that we get regarding the subject's body posture is more informative compared to the 2D information that we could get from the RGB video. The value of each pixel in a depth video stream frame is the distance, in millimeters, of the corresponding surface part of the object from the sensor.

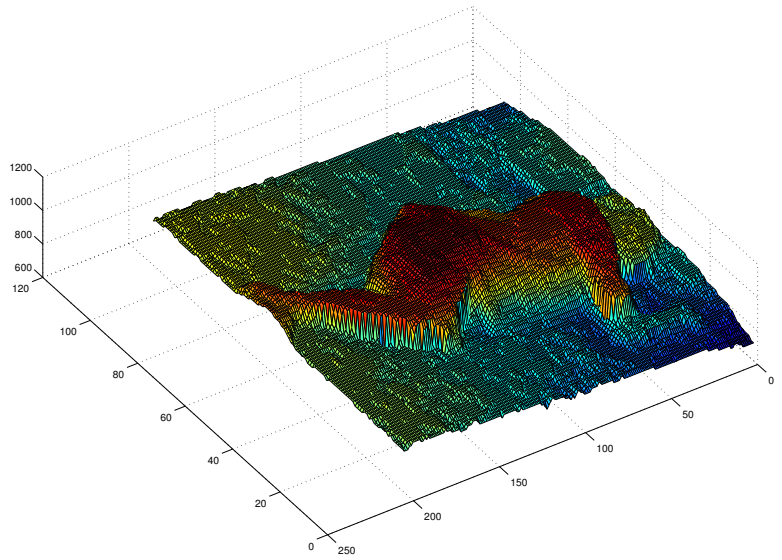


Figure 2.3. A 3D representation of the input obtained by the Kinect depth sensor.

2.2.2 Collection of Data to Monitor Medication Intake[3]

It is estimated that half the people taking prescription medication fail to stick to the regimen laid out by their doctor. As a solution to that problem, we have built the SmartDrawer, a medicine cabinet system that can track the usage of medication and prompt the user to remind them to take their prescription [10]. Benefits from such a system include increasing the quality of life for the patient, the ability to assist in the paperwork and other duties of a caregiver, and of course to verify information on drug consumption for research to study trends and effects. Such effects could be related to other cases of interest monitored at an assisted living home. For, example what are the effects of taking a particular medication to sleep quality, or what is the response of the patient to a given combination of drugs as opposed to a different combination.

[3]For more information about this project, the reader can refer to [10].

Radio Frequency Identification (RFID) is an emerging technology, being used in monitoring including healthcare. We apply different types of RFID tags to monitor drug taking and its impact in an assistive environment. Compared to other active Wireless Sensor Networks (WSNs), RFID tags do not need a battery, recharging, and so have no battery power loss problems. RFID tags are tiny in volume, and can be embedded into different objects. In this work we have built an RFID-based prototype application in an assistive environment called "Smart Drawer", which tracks medicine taking for the elderly. The system, not only provides reminders and alerts to the users but also logs their overall activity related to medication intake, which can be later used by caregivers or researchers to deduce useful conclusions regarding the effect of the medication to different individuals. Figure 2.4 presents an overview of the different components of the system.

2.2.3 Large-Scale Sensor Data Collection[4]

In contrast to what happens in cases the area to be monitored is limited and the locations and functionalities of the sensors to be used known in advance, in cases where we need to cover a larger area, like a whole apartment or a section of a hospital, there is very high uncertainty as to what sensors to use, where to place them and how to switch between active, power-save or other available modes. At the setting of an assisted living home for example, one can have static sensors which can cover one specific area, or mobile sensors, mounted for example at the waist of a human subject or at a mobile robotic platform, or there can be sensors that alter their area of coverage by panning and tilting. In such cases, to ensure optimal monitoring and data collection there is a need for an adaptive cooperative setup of sensors that can dynamically change to adapt to changes in the environment. To deal with such

[4]For more information about this project, the reader can refer to [11].

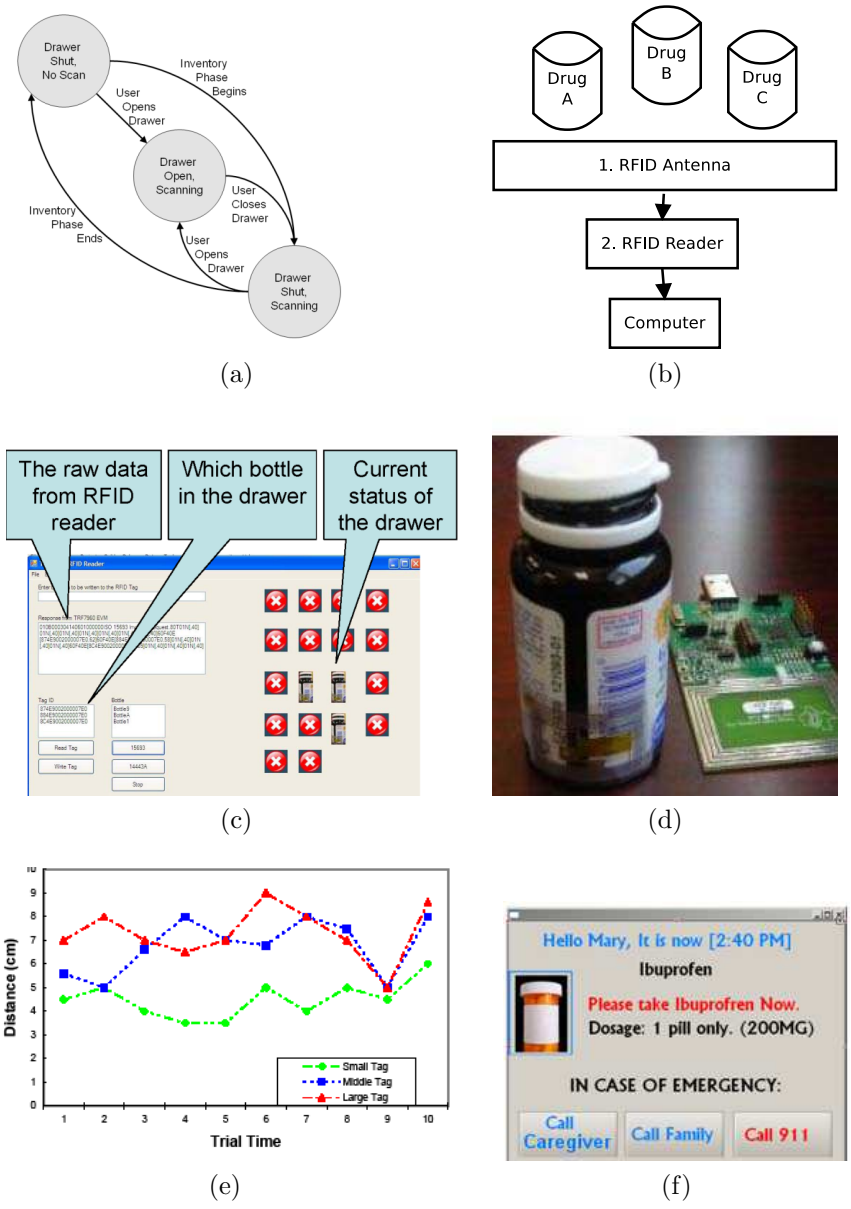


Figure 2.4. The smart drawer system. (a) State Machine of Smart Drawer Behavior. (b) Smart Drawer System Architecture. (c) Sample interface to be used by the administrator. (d) RFID reader and RFID tags on a bottle. (e) The maximum sensing distance between the different types of tags and the RFID reader. (f) Touch screen interface with sound alarms for the patient.

a challenge, we have proposed a methodology which allows for automated Sensor Placement and Coordination via Distributed Multi-Agent Cooperative Control [11].

The goal is to maximize the amount of information collected from the environment, given the limited amount of resources that the total of the available sensors can provide, and at the same time to be tolerant to failures of individual sensors by using a decentralized approach that re-organizes their placement in case of failures. We tackle this problem by employing a decentralized multi-agent coordination framework using message passing and the Max-Sum algorithm [12] for building and maintaining a common picture of the area to be monitored. We show that by representing each sensor as an independent agent which can take decisions individually and at the same time can affect the decisions of its neighboring sensor-agents we can provide a robust and efficient system for the monitoring of life-critical environments such as assistive environments or governmental infrastructures.

2.2.3.1 The Extended Max-Sum Decentralised Coordination Algorithm (EMSDC)

To deal with the problem of optimal placement, we created an extended version of the Max-Sum Decentralised Coordination (MSDC) [12] algorithm. It is a message passing algorithm applied on a factor graph. Factor graphs are graphical models that are used to represent functions of the form:

$$f(x_1, x_2, \ldots, x_n) = \prod_i \phi_i(X^i) \tag{2.1}$$

where X^i are subsets of $x_1, x_2, \ldots, x_n$ and $\phi_i(X^i) = p(x_i|parents(x_i))$.

A factor graph has two types of nodes. Variable nodes, that represent variables of the environment and factor nodes, that represent the factors $\phi_i(X^i)$. Edges are only allowed between variable nodes and factor nodes. For example, the distribution

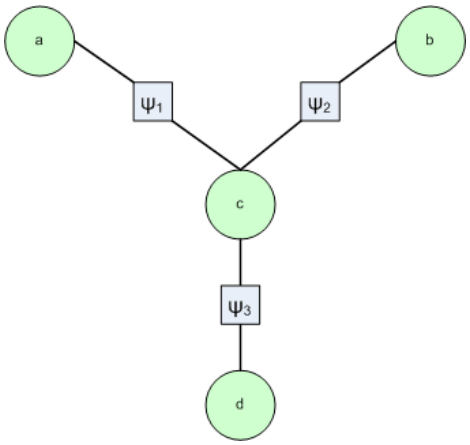


Figure 2.5. Example of a factor graph.

$p(a, b, c, d) = \psi_1(a, c)\psi_2(b, c)\psi_3(c, d)$ can be represented by the factor graph in Figure 2.5.

Note that we can have directed edges on factor graphs. Factor graphs with directed edges have the advantage that we can easily infer the assumed dependencies between variables.

EMSDC, unlike MSDC, takes into account not only the state of the agents but also the location of the agents (represented by utility - factor pairs). The main idea is that each agent has two types of states, task and location. This means that instead of having one factor graph, we have two, where at the second one variables represent agents' locations and utilities measure how good these locations are for each agent (typically a measure of the overlap with its neighbours multiplied by a gaussian function). We then run the MSDC two times, once for each factor graph, i.e. once for task selection and once for placement. As is the case with task selection, the agents exchange preferences on each other's location instead of their own actual location. Each agent then tries to push its neighbours away, to the direction that maximises each neighbour's utility. MSDC's performance has already been proven in [13]. Our

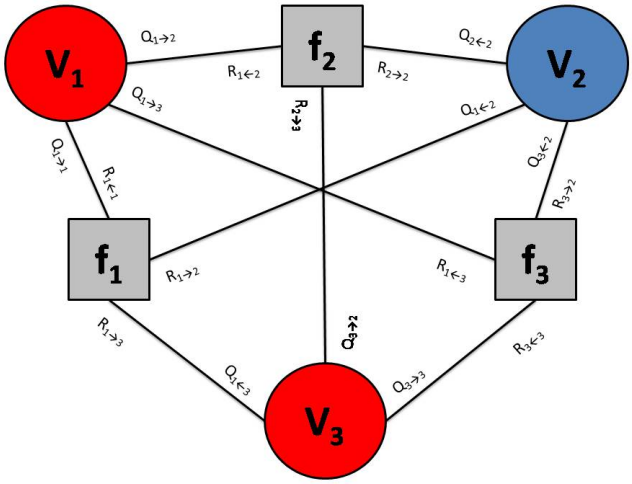


Figure 2.6. Example of messages being exchanged in the factor graph.

algorithm's running time is twice the running time of MSDC, but asymptotically the complexities are the same.

2.2.3.2 Optimal Placement with Gaussian Map

In most real world applications, when monitoring an area, there are important and not that important sections of that area. For example in an assistive living apartment, we probably do not want to monitor the inside of a closet or a storage area rarely used, and instead we want to focus on high traffic areas, such as the bathroom or the kitchen. In our model we represent this using a 3-dimensional Gaussian map.

For our experiments we used the Gaussian map (as viewed from above) depicted in Figure 2.7, and run the algorithm for 10 to 150 agents. Figure 2.7 shows the percentage of the covered area versus the number of agents. We can see that the total area covered rises rapidly in the lower dimensions and slower in the higher dimensions. This is because the agents are trying to cover the high interest areas (red) first, leaving others (blue) less covered. This might be a desirable feature, since

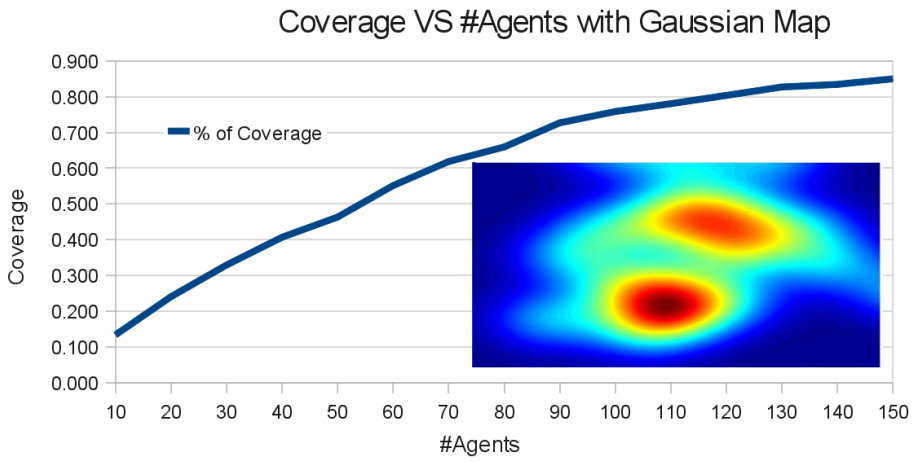


Figure 2.7. Percentage of coverage versus number of agents when using a Gaussian map.

we may have some overlap in the red areas, but this also means redundancy and increased fault tolerance. It is possible to tune the algorithm and put more weight on the overlap between the agents and less on the effect of the Gaussian map. This way we will have less overlap in the red areas and the agents will spread more.

2.2.3.3 System response to environmental changes

An interesting problem is how the system will respond to a change in the environment. To model this, we use two different Gaussian maps, depicted in Figure 2.8, where the second map (middle image) has one more "important" region. This could be an event like a fire in the kitchen or a person falling in the bathroom. The first image shows the initial random placement of the sensors, before the EMSDC algorithm has been run. The percentage of coverage in this case is 39.1% The second (middle) image shows the coverage of the area after the execution of the EMSDC algorithm but before the occurrence of the critical event. The coverage in that case is 68.5%. We then changed the map, at which point the coverage suddenly became

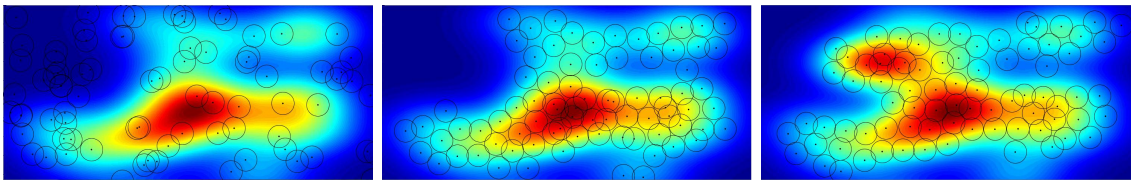


Figure 2.8. Gaussian map coverage. The left image shows a random starting state of the system when using a Gaussian map. The middle image shows the state of the system after the execution of the EMSDC algorithm. The right image shows the final state of the system after a change in the Gaussian map has taken place and the system has converged to a new solution.

60.7% since an important area was not covered. After the algorithm ran for 100 cycles the new resulting coverage increased to 64.2%. The rightmost image of Figure 2.8, shows the final position of the agents. We can clearly see that the agents adapt very well to the change in their environment. Note that it is not possible to achieve the initial percentage of coverage with the same number of sensors, since after the map change there is a bigger amount of "important" regions to be covered.

2.2.3.4 Fault Tolerance

The two main benefits of using multi agent systems are decentralised control, meaning that each agent performs small tasks that can be performed by low cost devices, and fault tolerance. Here we prove that EMSDC performs very well in the presence of failures.

To test the fault tolerance of the system, we compared it to a static system, i.e. a system where the sensors cannot move to compensate for failures. We run EMSDC using 100 agents, calculated the coverage after 0% to 40% agents have failed randomly and compared the results with the coverage of EMSDC in the presence of failures. To simulate failures, we use a model where each sensor has a probability p to fail at each cycle. After that point the sensor becomes useless either because it

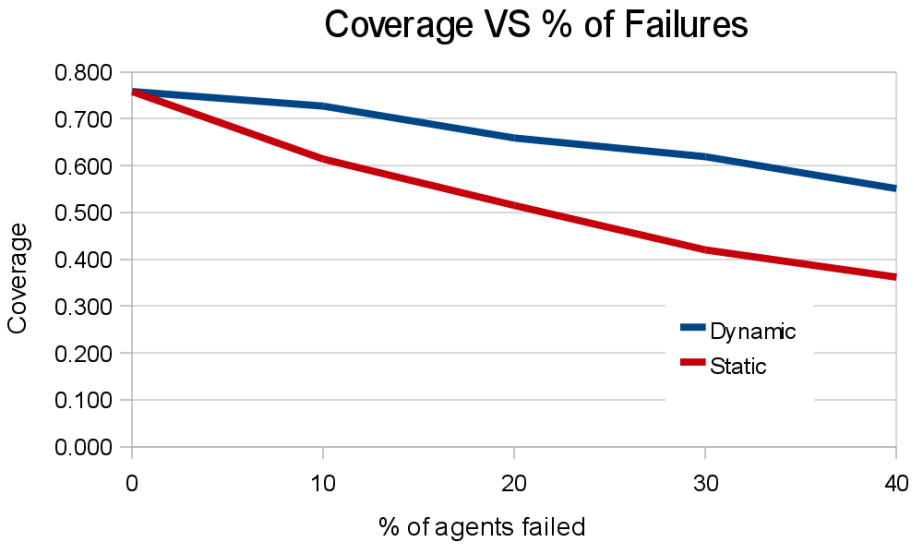


Figure 2.9. Percentage of coverage versus percentage of agents failed, for a static and dynamic system. We can see that the dynamic system adapts well to failures.

cannot take measurements or because it cannot transmit/receive data. We can see the results in Figure 2.9. The very small decrease in coverage for EMSDC means that as the agents in important sections fail, others come and take their place. Contrary if the placement of the sensors could not be re-organized after some sensors have failed, important regions could remain uncovered and that would result in a rapid decrease in percentage of coverage. As we can see from the graph, with 10% of sensors failed we already have a 13% difference in the coverage between the static approach and our dynamic system.

2.2.4 Higher Level Data Collection[5]

When dealing with data coming from a number of different sensors simultaneously, sometimes it does not make sense to examine each data source separately. The collected data coming from individual sources are only useful if they are examined

[5]For more information about this work, the reader can refer to [14].

longitudinally and are correlated temporally or spatially with data collected from other sources. For example, if would like to determine the trajectory of a person going from their bed to their refrigerator which is located in another room, we will need to examine all the sensors that were triggered along the path followed by the person from the source to the destination. Moreover, in many cases we are not interested in logging all the sequences of events that occur over time, but only those which would require our attention. We call each individual sensor activation an "event" and each sequence of events and "episode". Our goal is to identify and log abnormal episodes, or in other words, episodes that would require our attention.

In this section, we suggest a method [14] that detects abnormal behavior using wireless sensor networks. We model an episode as a series of events, which includes spatial and temporal information about the subject being monitored. We define a similarity scoring function that compares two episodes taking into consideration temporal aspects. To determine if an episode is abnormal or not we compare it to a database of predefined normal and abnormal episodes. We propose a way to determine the threshold to divide episodes into two groups that minimizes wrong classification. Weights on individual functions that consist the similarity function are determined experimentally so that they can produce the best results in terms of area under curve in receiver operating characteristic (ROC) curve.

2.2.4.1 Definitions

An *event* is a 3-tuple which includes a sensor ID, a time stamp, and a duration. We let e_i be an event, where i indicates the order of activated sensors.

$$e_i = (S, T, D) \tag{2.2}$$

where S is a sensor ID that can represent the location of the sensor or an individual action, T is a time stamp when the sensor is activated, and D is a duration, which is time difference after one sensor is activated until the next sensor is activated.

An *episode* is a series of events. We let E_i be an episode, where i indicates the index and define it as a sequence.

$$E_i = (e_1, e_2, ..., e_n) \tag{2.3}$$

The order of events in an episode is determined by the timestamp T of e_i. For example, when a person walks from a bedroom to a kitchen through a hallway, three sensors may react by detecting change of light intensity. In this case, we have three events, e_1, e_2, and e_3, which are corresponding to a sensor at a bedroom, a sensor on a hallway, and a sensor at a kitchen, respectively.

We define *abnormal behavior* as "an episode which has not occurred before at all, an episode which was rarely occurred before, or an episode which was not close enough to any of the ones that have previously occurred." But this is not enough to define abnormal behavior since we do not consider temporal aspects in episodes. First, we need to consider time and add it to the definition that "an episode whose sequence of events are similar to the previous one, but the time of the day that the episode happened is very different from the previous one." Second, we need to consider the duration of each event. Same sequences of events that happened at similar times can have different duration. An example includes that a person goes to a bathroom at 1:00 am, and usually stays less than 10 minutes, but if the same person stays at the bathroom for longer time, it should be regarded as an abnormal behavior. Therefore, we need to add it to the definition that "an episode whose sequence of events are

similar and whose time it happened is close to the previous one, but whose duration for each event is not close enough to the previous one."

To handle each of the above cases and reach a final consensus as to if an episode is abnormal or not we define a set of different similarity sub-functions s_i for each case, and then we combine them to a global similarity function S by giving them appropriate weights.

$$S(E_1, E_2) = \sum_{i}^{n} w_i s_i \tag{2.4}$$

where, E_1 and E_2 are arbitrary episodes, whose lengths are the same, w_i is the weight, and s_i is an individual similarity measuring function. Every s_i is normalized so that it can have a value between 0 and 1. The final decision if an episode is abnormal or not is based on that score. For more information about how the weights w_i, and the individual similarity functions s_i are calculated, as well as the experimental results regarding the effectiveness of this method, there reader can refer to our work in [14].

2.2.5 Ensuring the Quality of the Collected Data[6]

Building a large-scale sensor network of a set of heterogeneous sensing devices can pose serious challenges with regard to the processing and storage of the generated data. Especially in a dynamically changing network where sensors, can be activated or added according to the temporary, local needs, the amount of data generated a certain parts of the network can be unpredictable. In such cases, if the processing or transmission capacity of network is exceeded, data can be lost or the system can completely fail. To facilitate for such special situations and ensure the quality of the data collected by the system, we have proposed the use of an Ontology Centered Middleware which will handle the cooperation among the different devices or appli-

[6]For more information about this work, the reader can refer to [15].

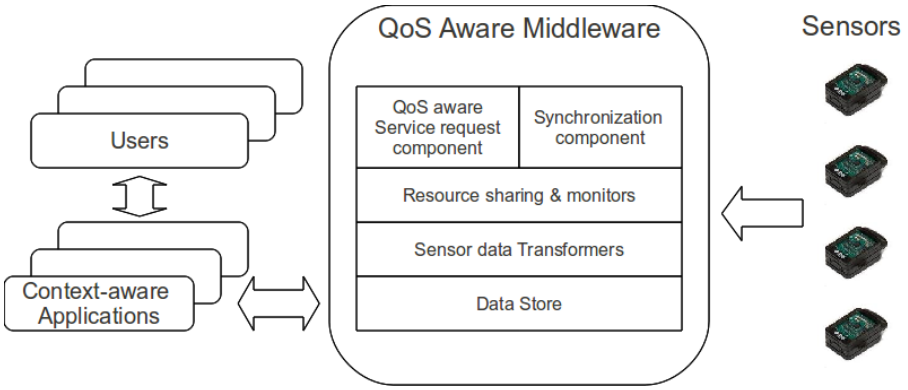


Figure 2.10. An example ontology centered middleware architecture.

cations running in the network [15]. Our goal is to maximize the utilization of the data generated by the network while at the same time providing quality of service, to avoid data loss or unacceptable delays in case of real-time applications. In this context, the ontologies are used as a means of representing and exchanging specifications regarding client requirements about data generation and transmission rates, queries and expected response times from other client/server applications, as well as for high-level data transmission. Figure 2.10 gives an illustration of the architecture of the proposed middleware.

In order to allow an ontology centered middleware architecture to provide QoS support to context-aware applications, we need to provide an infrastructure to allow applications to describe their structure and query patterns; this is generally referred to as QoS specification. We characterize an application as ultimately consisting of queries, which have end-to-end delay requirements. There are many factors that influence the end-to-end delay of an application's queries. Most of them can be handled through heuristics and multi-resource reservation, such as those to manage network bandwidth, memory usage, task scheduling, and I/O. However, ontology centered middleware requires the use of an inference engine, where it is not possible

to determine the inference time unless the size of the data set used is known. In order to know how much data will be used by a query, it is necessary to establish restrictions on how data is generated for the ontologies, and their corresponding properties. Each property in an ontology can have a restriction on how many data entries can be associated with that property. We call this the cardinality of a property. This posses a conflict of interest, as the data restrictions that are necessary for one context aware application might not be suitable for another context-aware application. A possible solution would be to have both applications rely on a different set of ontologies with different cardinality constraints for their properties, but that would defeat the purpose of an ontology centered middleware architecture, whose greatest value is a unified model for knowledge and data representation. To solve this problem we propose a trade-off where we relax the unity of the data in order to allow some level of QoS support. This component allows the client to have some level of participation on the process of converting raw data into ontological data. This is done in order for different applications to be able to modify the same sensor data and produce different data while storing it using the same knowledge representation. While this might seem counter-intuitive, the goal is to make a fine-grained distinction in the different ontological properties used by the applications, where a single property can be treated as a set of different ¡property, cardinality¿ couples by the middleware's inference engine. The challenge is to make this process completely transparent to the client application. The basic flow of our proposed middleware is shown in 2.11. For more information about the proposed architecture the reader to our work in [15].

2.3 Collection of Human Genomic and Physiological Data

Traditionally, in order to understand the functionality of the human body and deal with possible abnormalities, the physicians have to examine data coming from the

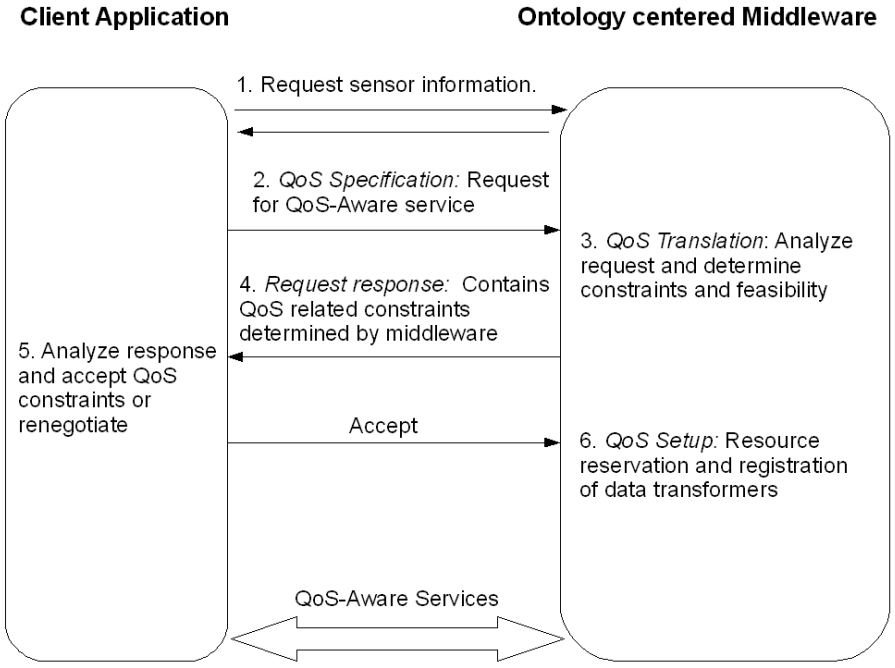


Figure 2.11. The proposed QoS negotiation mechanism.

human physiology. Human physiology [16] is the science of the mechanical, physical, bioelectrical, and biochemical functions of humans in good health, their organs, and the cells of which they are composed. In recent years, there has been a shift of focus towards genomics and their effect in human physiology and behavior. There is strong evidence that many diseases or physiological abnormalities originate in genomic abnormalities or alterations. Researchers have tried to discover correlations between genomic and physiological abnormalities and combine these two different sources of information for better diagnosis, prognosis and disease treatment. In other words, there have been research efforts to connect the human genotype with the phenotype[7].

[7]According to Medterms.com, *Genotype* is the genomic constitution (the genome) of a cell, an individual or an organism. The genotype is distinct from its expressed features, or phenotype. The genotype of a person is her or his genomic makeup. It can pertain to all genes or to a specific gene. By contrast, the *Phenotype* results from the interaction between the genotype and the environment. It is the composite of the characteristics shown by the cell, individual or organism under a particular set of environmental conditions.

In this work we will show how genotypic data can be used together with physiological/phenotypic data for disease prognosis and treatment monitoring. Furthermore, we will see that the same computational methods that we have used to analyze behavioral data, can be also used to analyze genomic and physiological data, thus creating a common framework to analyze different modalities of data coming from the same human subjects. In most cases the raw data obtained by medical measurements are noisy and redundant. In addition, there is no obvious way of directly using the data to extract information regarding the examined disease. In the next chapters we will see methods to extract features that can be used for disease classification and select the most important features related to the disease. To reduce noise and avoid over-fitting a feature selection step is necessary before training and classification. An extra advantage of the feature selection process is that the majority of the irrelevant features are discarded and the few remaining can be indicators of possible biomarkers related to the observed disease.

The collection of genomic and physiological data in most cases requires special medical laboratory equipment, therefore it is out of the scope of this work to propose new methods for doing so. However, since in this work we propose methods for processing and analyzing such data, in the next sub-sections, we will give a brief description of the data that we used in our experiments and the method that is usually used by physicians to collect such data. Our experiments focus on methods for cancer diagnosis, prognosis and progression monitoring. Following we present the data format and collection methods of cancer-related data.

2.3.1 HRMAS ^{1}H MRS Data

Magnetic resonance spectroscopic (MRS) studies of brain biomarkers can provide statistically significant biomarkers for tumor grade differentiation and improved

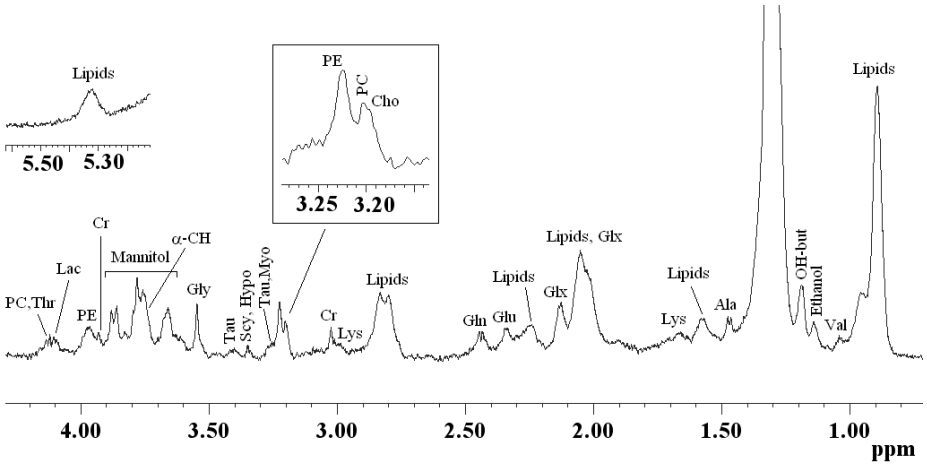


Figure 2.12. *Ex vivo* HRMAS ^{1}H MR spectrum of a 5.8 mg glioblastoma multiforme (GBM) tissue biopsy. Val, Valine; OH-but, OH-butyrate; Lac, Lactate; Ala, Alanine; Lys, Lysine; Glx, β-CH2 of Glutamine and Glutamate; Glu, Glutamate; Gln, Glutamine; Cr, Creatine; Tau, Taurine; Myo, Myo-inositol; Hypo, Hypotaurine; Scy, Scyllo-inositol; Gly, Glycine; β-CH of aliphatic aminoacids; PE, PhosphoEtanolamine; Thr, Threonine; PC, PhoshoCholine; Cho, Choline. The insert shows the choline containing compounds region.

predictors of cancer patient survival [17]. Ex vivo high-resolution magic angle spinning HRMAS proton ^{1}H MRS of unprocessed tissue samples can help interpret in vivo ^{1}H MRS results, to improve the analysis of micro-heterogeneity in high-grade tumors [18]. Furthermore, two-dimensional HRMAS ^{1}H MRS enables more detailed and unequivocal assignments of biologically important metabolites in intact tissue samples [19]. In Figure 2.12, an ex vivo HRMAS ^{1}H MR spectrum of a 1.9 mg anaplastic ganglioglioma tissue biopsy is shown together with metabolites values that correspond to each frequency of the spectrum.

2.3.2 Gene Expression Data

According to MedTerms.com[8], Gene Expression, is the translation of information encoded in a gene into protein or RNA. Expressed genes include genes that are transcribed into messenger RNA (mRNA) and then translated into protein, as well as genes that are transcribed into types of RNA such as transfer RNA (tRNA) and ribosomal RNA (rRNA) that are not translated into protein. Gene expression is a highly specific process in which a gene is switched on at a certain time and "speaks out." Figure 2.13[9] shows an example of how the double helix DNA is transcribed into RNA and how later the RNA is translated into proteins which control the functions of the cell.

A major focus in cancer research is to identify genes, using DNA-microarrays, that are aberrantly expressed in tumor cells, and to use their aberrant expression as biomarkers that correspond to and facilitate precise diagnoses and/or therapy outcomes of malignant transformation [20]. In our study, the Affymetrix gene-chip U133Plus®DNA microarray of the complete human genome was used to perform transcriptome profiling on each specimen. The raw expression data were analyzed for probe intensities using the Affymetrix GeneChip expression analysis manual procedures; and the data were normalized using current R implementations of RMA algorithms [21].

2.3.3 Array Comparative Gene Hybridization (aCGH) Data

Array comparative genomic hybridization (aCGH) is a recently introduced technique for identifying chromosomal aberrations in human diseases throughout the human genome. aCGH can be used for detection and mapping of copy number abnor-

[8]http://www.medterms.com/script/main/art.asp?articlekey=3564
[9]Image borrowed from Wikipedia.org.

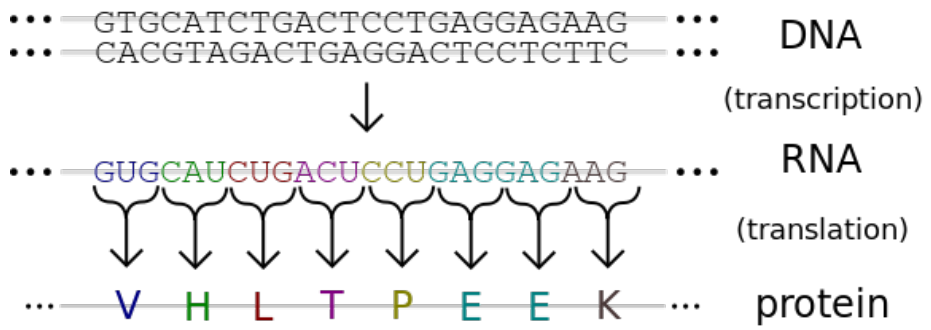


Figure 2.13. An example of how Genes are expressed by being transcribed into RNA, and subsequently translated into proteins.

malities which can be associated with certain disease phenotypes. Specific patterns in DNA copy number variations (CNVs) can be associated with certain disease types and can facilitate prognosis and progress monitoring of the the disease. This, in turn, can facilitate the localization of critical genes related to specific diseases which can be used as biomarkers for disease diagnosis, prognosis and response to therapy [22, 23].

A set of chromosomal aberrations occurring consistently when a certain disease is observed can indicate that there is correlation between those aberrations and the observed disease. Such patterns have been utilized by researchers [24, 25, 26, 27, 28, 22, 29, 30, 31, 32, 33] for cancer detection and typing. In general, the number of probes of a high-resolution CGH can span from hundreds to thousands. Contrary, only a few genes are associated with most diseases.

Figure 2.14 visualizes a cancerous sample which contains colorectal cancer with liver metastasis. In 2.14b we can see the original log-ratios of the DNA copy number variations throughout the chromosome. In 2.14c we can see the pointwise averaging of all computed profiles after the sample has been segmented. During segmentation, each single-sample signal is divided into regions of constant copy number, called segments [26, 34]. Finally, 2.14d shows 4 different heatmaps obtained from the same sample.

The first line is the heatmap of the original log-ratios; the last is the heatmap of the averaged profile (pointwise averaging across the outputs of all algorithms); and the lines in the middle are the heatmaps corresponding to the data discretized and smoothed by different algorithms (CBS [35], CGHseg [31] and cghFLasso [36]).

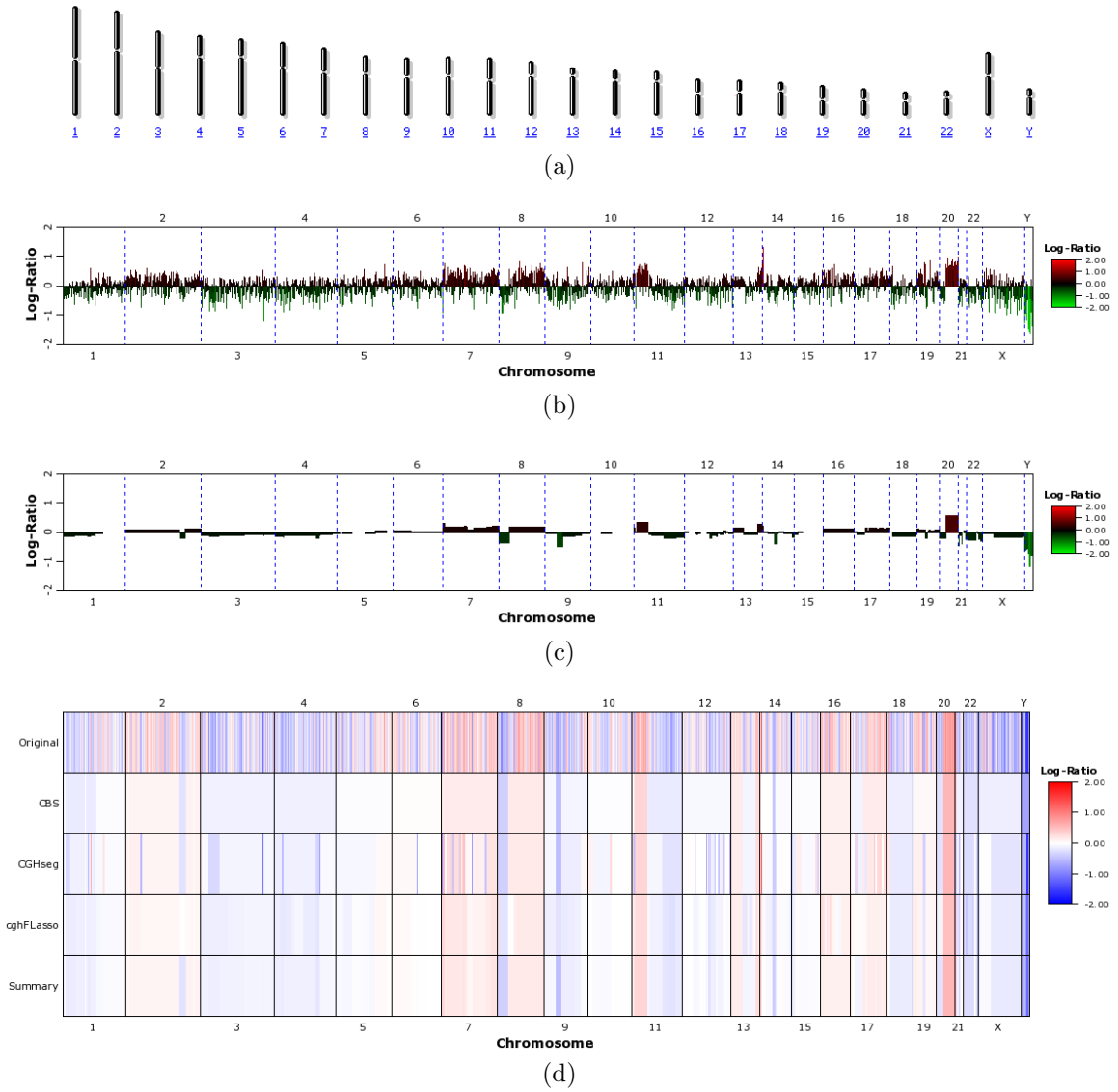


Figure 2.14. The images in this figure visualize the CNVs of a sample of colorectal cancer with liver metastasis. (a) Full male human genome. (b) Original data. Chromosome numbers are given on top and bottom of the image. Log-ratios are indicated by both the y-axis and the color (green indicates regions of chromosomal loss and red indicates regions of chromosomal gain). (c) Summary data (Pointwise averaging of all computed profiles). (d) CNV Heatmap. The first line is the heatmap of the original log-ratios; the last is the heatmap of the averaged profile (pointwise averaging across the outputs of all algorithms); and the lines in the middle are the heatmaps corresponding to the data discretized and smoothed by different algorithms (CBS [35], CGHseg [31] and cghFLasso [36]). To visualize the data we used the CGHweb tool (http://compbio.med.harvard.edu/CGHweb/).

CHAPTER 3

EFFICIENT FEATURE SELECTION IN HUMAN-CENTERED DATA

3.1 Introduction

The data collected from human subjects can be used in several ways in order to extract useful information regarding the well-being of the examined subject. Machine Learning is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. The advantage of Machine Learning lies int he fact that it allows us to analyze huge datasets and extract knowledge with minimal human intervention. In the general case, machine learning algorithms base their decisions on the properties of a set of features regarding the analyzed data. These features can either be discovered automatically or defined manually. However, in most cases, out of the big set of features that can be extracted from a dataset, only a small subset is related to the examined problem. Identifying that subset is crucial to the success of the outcome of the application. Using the full set of available features, not only increases the complexity of the problem and requires more computational resources, but also decreases the accuracy of the result due to added noise.

In this chapter we attempt to tackle that problem by proposing a method to identify those features which are the most related to our problem. These methods are known as Feature Selection methods. A variety of feature selection methods have been proposed in the past [37]. What differentiates our proposed method from existing ones is that it is more suited to human-centered data and it gives better accuracy when used for Supervised Learning compared to other methods. In the

following sections of this chapter we present our proposed feature selection method and show its superiority when applied to array comparative genomic hybridization (aCGH) data.

3.2 Hybrid Sparsity Regularization (HSR) for Feature Selection in aCGH Data

Array comparative genomic hybridization (aCGH) is a newly introduced method for the detection of copy number abnormalities associated with human diseases with special focus on cancer. Specific patterns in DNA Copy Number Variations (CNVs) can be associated with certain disease types and can facilitate prognosis and progress monitoring of the the disease. Machine learning techniques can be used to model the problem of tissue typing as a classification problem. Feature selection is an important part of the classification process and several feature selection methods have been examined in the different domains where classification has been applied. In this work we present a novel feature selection method based on sparsity regularization which shows a promising performance when used for classification of aCGH data. To validate the performance of the proposed method we experimentally compare it with existing feature selection methods on four publicly available aCGH datasets.

Chromosomal aberrations occur in many diseases. For example, in cancer, increases or decreases in DNA copy number can alter the expression levels of tumor suppressor genes and oncogenes resulting in tumor genesis. Array comparative genomic hybridization (aCGH) is a recently introduced technique for identifying chromosomal aberrations in human diseases throughout the human genome. aCGH can be used for detection and mapping of copy number abnormalities which can be associated with certain disease phenotypes. This, in turn, can facilitate the localization of critical genes related to specific diseases which can be used as biomarkers for disease diagnosis, prognosis and response to therapy [22, 23].

Machine Learning techniques can be used to discover patterns in DNA copy number variations associated with certain diseases. A set of chromosomal aberrations occurring consistently when a certain disease is observed can indicate that there is correlation between those aberrations and the observed disease. Such patterns have been utilized by researchers [24, 25, 26, 27, 28, 22, 29, 30, 31, 32, 33] for cancer detection and typing. In the general case, the task to accomplish is the classification of tissue samples as cancerous or non-cancerous, and extensively their classification to a specific cancer type.

In the setting of supervised learning, the copy number changes of particular locations (probes) of the genome are used as features for training and classification. In general, the number of probes of a high-resolution CGH can span from hundreds to thousands. Contrary, only a few genes are associated with most diseases. Moreover, the number of available samples to be used for training is usually only a few dozens. To reduce noise and avoid over-fitting a feature selection step is necessary before training and classification. An extra advantage of the feature selection process is that the majority of the irrelevant features are discarded and the few remaining can be indicators of possible biomarkers related to the observed disease.

Feature selection has already been shown to significantly benefit the classification accuracy of aCGH data [26, 27, 30]. In this work we introduce a novel feature selection method based on sparsity regularization that produces higher accuracy compared to the methods that have been previously tested on aCGH data. Our method is inspired by multi-task learning and feature selection [38, 39], which have developed a similar model of $\ell_{2,1}$-norm regularization to couple feature selection across tasks. Previous works have examined sparsity regularization in dimensionality reduction and feature selection [40, 41]. ℓ_1-norm regularization can be used by regression or SVM models to perform feature selection by shrinking the coefficients of the irrele-

vant features to zero. However, the number of features that can be selected by this method is bounded by the number of the samples in the training dataset. ℓ_2-norm regularization does not have that limitation but is sensitive to outliers and results in decreased classification accuracy when used. Nie et.al. [42] proposed the use of joint $\ell_{2,1}$-norm minimization on both loss function and regularization. Unlike ℓ_2-norm which is sensitive to outliers, $\ell_{2,1}$-norm can effectively remove outlying values. In addition, a $\ell_{2,1}$-norm is performed to select features across all data points with joint sparsity. That means that each feature has small scores for all or has large scores over all data points.

In this work we propose a hybrid regularization method which uses two separate regularization terms involving $\ell_{2,1}$-norm and ℓ_1-norm. This is particularly important in multi-class classification problems which contain a big number of classes because a feature, for example, that is important for one class but not important for all others get a low total score (coefficient) a be lost in the feature selection process. Our method ensures that such features will at least get a high coefficient value for the classes that they are important to and have more chances to be included in the final set of selected features. Each regularization term is assigned a different weight according to the specifics of the dataset. We also propose an efficient algorithm to solve the objective function of our method.

To test the performance of our proposed method we conducted experiments on four different, publicly available aCGH datasets. We compare with other methods that have been recently proposed for feature selection on aCGH data and present the classification accuracy results using SVM [43] and Logistic Regression [44, 45] as classifiers.

3.3 Feature Selection Methodology

3.3.1 Hybrid Sparsity Regularization (HSR)

Feature selection methods can be divided into wrappers, filters and embedded methods. Wrapper methods utilize the learning machine of interest as a black box to select the subset of features that give the best predictive accuracy. Filter methods select features based on discriminant criteria that rely on the characteristics of data, independent of any classification algorithm. Filter methods are limited in scoring the predictive power of combined features, and thus have shown to be less powerful in predictive accuracy as compared to wrapper methods, whereas wrapper methods are much slower and cannot be efficiently applied to large datasets. Embedded methods perform feature selection as part of the training process and are usually specific to given learning machines [46].

In this work we will introduce a filter feature selection method based on least square regression with ℓ_2-norm minimization on the loss function and hybrid $\ell_{2,1}$-norm and ℓ_1-norm regularization.

Least square regression has been widely used for classification. Given training data $\{x_1, x_2, \dots x_n\} \in \mathbb{R}^d$ and the associated class labels $\{y_1, y_2, \dots y_n\} \in \mathbb{R}^c$, traditional least square regression solves the following optimization problem to obtain the projection matrix $W \in \mathbb{R}^{d \times c}$ and the bias $b \in \mathbb{R}^c$:

$$\min_{W,b} \sum_{i=1}^{n} \|W^T x_i + b - y_i\|_2^2. \tag{3.1}$$

For simplicity, the bias b can be absorbed into W when the constant value 1 is added as an additional dimension for each data point $x_i, (1 \leq i \leq n)$. Thus the problem becomes:

$$\min_{W,b} \sum_{i=1}^{n} \|W^T x_i - y_i\|_2^2. \tag{3.2}$$

To control variance and prevent overfitting we need to add one or more regularization terms to the above equation. Several regularizations are possible:

$$R_1(W) = \|W\|^2,$$

$$R_2(W) = \sum_{j=1}^{c} \|w_j\|_1,$$

$$R_3(W) = \sum_{i=1}^{d} \|w^i\|_2^0,$$

$$R_4(W) = \sum_{i=1}^{d} \|w^i\|_2.$$

$R_1(W)$ is the ridge regularization which suffers from the existence of outliers in the dataset due to high variance. $R_2(W)$ is the LASSO regularization which has the desired property of giving different weights to a feature across different classes c but produces very sparse solutions, especially when the number of samples is small. $R_3(W)$ and $R_4(W)$ penalize all c regression coefficients corresponding to a single feature as a whole. Although the ℓ_0-norm of $R_3(W)$ is the most desirable [47], it is difficult to compute, so in this work we use $R_4(W)$ instead which gives similar results, in combination with $R_2(W)$, thus creating a hybrid regularization term which combines the desired properties of both while reducing the non-desired properties of each at the same time.

By adding the two regularization terms our problem becomes:

$$\min_W J(W) = \sum_{i=1}^{n} \|W^T x_i - y_i\|^2 + \gamma_1 R_2(W) + \gamma_2 R_4(W) \tag{3.3}$$

or

$$\min_W J(W) = \|W^T x_i - y_i\|^2 + \gamma_1 \|W\|_1 + \gamma_2 \|W\|_{2,1}. \tag{3.4}$$

Although solving this problem seems difficult as all terms are non-smooth, we will show in the next section that is can be efficiently solved. For short we will call this objective function "L2R21R2". The optimal value of the parameters γ_1 and γ_2 can be determined experimentally from the dataset. The resulting values in the projection matrix W will determine the optimal coefficient values for each attribute x_i. To select the best k features we can just sort the features by decreasing coefficient value and keep the top k of them. Figure 3.1 shows a visualization of the coefficient table W after the application of HSR feature selection method on aCGH dataset 3 (see section 3.4). In the visualized gray-scale heat-map, each row represents a class and each column represents a feature. The gray-scale color of each square represents the calculated coefficient value of the feature for the corresponding class. Lighter color means the coefficient has a positive value, darker color means negative coefficient value, gray color means a value close to 0. Large absolute values for each square indicate strong correlation for the corresponding feature-class pair. The overall importance of each feature is measured by calculating the sum of the absolute values of the feature for all classes. In the figure, the features are sorted from left to right by total importance value.

Figure 3.1. Visualization of the coefficient table W after the application of HSR feature selection method on aCGH dataset 3. Each row represents a class, each column represents a feature. The grayscale color of each square represents the final coefficient value of the feature for the corresponding class. Lighter color means the coefficient has a positive value, darker color means negative coefficient value, gray color means a value close to 0. The features are sorted from left to right by total importance value.

3.3.2 An Efficient Algorithm to Solve L2R21R2

Although our objective function is convex, it is difficult to be solved. Because both regularization terms are non-smooth. It was generally felt that the $\ell_{2,1}$-norm minimization problem is much more difficult to solve than the ℓ_1-norm minimization problem. Existing algorithms usually reformulate it as a second-order cone programming (SOCP) or semidefinite programming (SDP) problem, which can be solved by interior point method or the bundle method. However, solving SOCP or SDP is computationally very expensive, which limits their use in practice. Here, we propose an efficient algorithm to solve our objective function in Eq. (3.4).

The Eq. (3.4) can be written as:

$$\min_W Tr(X^T W - Y)^T (X^T W - Y) + \gamma_1 \|W\|_1 + \gamma_2 \|W\|_{2,1}. \tag{3.5}$$

Taking the derivative w.r.t $w_i (1 \leq i \leq c)$, and setting it to zero, we have

$$XX^T w_i - Xy_i + \gamma_1 D_i w_i + \gamma_2 \tilde{D} w_i = 0, \tag{3.6}$$

where $D_i(1 \leq i \leq c)$ is a diagonal matrix with the k-th diagonal element as $\frac{1}{2|w_{ki}|}$, $\tilde{D}$ is a diagonal matrix with the k-th diagonal element as $\frac{1}{2\|w^k\|_2}$. Thus,

$$w_i = (XX^T + \gamma_1 D_i + \gamma_2 \tilde{D})^{-1} X y_i. \tag{3.7}$$

Note that D_i and $\tilde{D}$ depend on W and thus is also unknown variables. We propose an iterative algorithm to solve this problem, and the algorithm is described in Algorithm 1.

Input: X, Y
Initialize $W^1 \in \mathbb{R}^{d \times c}$, $t = 1$;
while *not converge* **do**
1. Calculate the diagonal matrices $D_i^{(t)}(1 \leq i \leq c)$ and $\tilde{D}^{(t)}$, where the k-th diagonal element of $D_i^{(t)}$ is $\frac{1}{2|w_{ki}^{(t)}|}$, the k-th diagonal element of $\tilde{D}^{(t)}$ is $\frac{1}{2\|(w^{(t)})^k\|_2}$;
2. For each $i(1 \leq i \leq c)$, $w_i^{(t+1)} = (XX^T + \gamma_1 D_i^{(t)} + \gamma_2 \tilde{D}^{(t)})^{-1} X y_i$;
3. $t = t + 1$;

end
Output: $W^{(t)} \in \mathbb{R}^{d \times c}$.

Algorithm 1: Hybrid Sparsity Regularization Algorithm

3.3.3 Algorithm Analysis

We will prove that the above algorithm converges to the global optimum.

Lemma 3.3.1 $\|w\|_2 - \frac{\|w\|_2^2}{2\|w_0\|_2} \leq \|w_0\|_2 - \frac{\|w_0\|_2^2}{2\|w_0\|_2}$

Proof: Obviously, $-(\|w\|_2 - \|w_0\|_2)^2 \leq 0$, thus we have

$$-(\|w\|_2 - \|w_0\|_2)^2 \leq 0 \tag{3.8}$$

$$\Rightarrow 2\|w\|_2 \|w_0\|_2 - \|w\|_2^2 \leq \|w_0\|_2^2 \tag{3.9}$$

$$\Rightarrow \|w\|_2 - \frac{\|w\|_2^2}{2\|w_0\|_2} \leq \|w_0\|_2 - \frac{\|w_0\|_2^2}{2\|w_0\|_2} \tag{3.10}$$

which completes the proof. □

Theorem 3.3.2 *The algorithm decreases the objective value in each iteration.*

Proof: According to Step 2 in the algorithm, we have

$$\begin{aligned} W^{(t+1)} = & \\ & \min_{W} Tr(X^T W - Y)^T (X^T W - Y) \\ & + \gamma_1 \sum_{i=1}^{c} w_i^T D_i^{(t)} w_i + \gamma_2 Tr W^T \tilde{D}^{(t)} W, \end{aligned} \tag{3.11}$$

therefore we have

$$\begin{aligned} & Tr(X^T W^{(t+1)} - Y)^T (X^T W^{(t+1)} - Y) \\ & + \gamma_1 \sum_{i=1}^{c} (w_i^{(t+1)})^T D_i^{(t)} w_i^{(t+1)} + \gamma_2 Tr(W^{(t+1)})^T \tilde{D}^t W^{(t+1)} \\ & \leq Tr(X^T W^{(t)} - Y)^T (X^T W^{(t)} - Y) \\ & + \gamma_1 \sum_{i=1}^{c} (w_i^{(t)})^T D_i^{(t)} w_i^{(t)} + \gamma_2 Tr(W^{(t)})^T \tilde{D}^{(t)} W^{(t)} \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad & Tr(X^T W^{(t+1)} - Y)^T (X^T W^{(t+1)} - Y) \\ & + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \frac{(w_{ij}^{(t+1)})^2}{2 \left| w_{ij}^{(t)} \right|} + \gamma_2 \sum_{k=1}^{d} \frac{\left\| (w^{(t+1)})^k \right\|_2^2}{2 \left\| (w^{(t)})^k \right\|_2} \\ & \leq Tr(X^T W^{(t)} - Y)^T (X^T W^{(t)} - Y) \\ & + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \frac{(w_{ij}^{(t)})^2}{2 \left| w_{ij}^{(t)} \right|} + \gamma_2 \sum_{k=1}^{d} \frac{\left\| (w^{(t)})^k \right\|_2^2}{2 \left\| (w^{(t)})^k \right\|_2} \end{aligned}$$

$$
\begin{aligned}
\Rightarrow \quad & Tr(X^T W^{(t+1)} - Y)^T (X^T W^{(t+1)} - Y) \\
& + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \left(\frac{(w_{ij}^{(t+1)})^2}{2\left|w_{ij}^{(t)}\right|} - \left|w_{ij}^{(t+1)}\right| + \left|w_{ij}^{(t+1)}\right| \right) \\
& + \gamma_2 \sum_{k=1}^{d} \left(\frac{\left\|(w^{(t+1)})^k\right\|_2^2}{2\left\|(w^{(t)})^k\right\|_2} - \left\|(w^{(t+1)})^k\right\|_2 + \left\|(w^{(t+1)})^k\right\|_2 \right) \\
& \leq Tr(X^T W^{(t)} - Y)^T (X^T W^{(t)} - Y) \\
& + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \left(\left|w_{ij}^{(t)}\right| + \frac{(w_{ij}^{(t)})^2}{2\left|w_{ij}^{(t+1)}\right|} - \left|w_{ij}^{(t)}\right| \right) \\
& + \gamma_2 \sum_{k=1}^{d} \left(\left\|(w^{(t)})^k\right\|_2 + \frac{\left\|(w^{(t)})^k\right\|_2^2}{2\left\|(w^{(t)})^k\right\|_2} - \left\|(w^{(t)})^k\right\|_2 \right)
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad & Tr(X^T W^{(t+1)} - Y)^T (X^T W^{(t+1)} - Y) \\
& + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \left|w_{ij}^{(t+1)}\right| + \gamma_2 \sum_{k=1}^{d} \left\|(w^{(t+1)})^k\right\|_2 \\
& \leq Tr(X^T W^{(t)} - Y)^T (X^T W^{(t)} - Y) \\
& + \gamma_1 \sum_{i=1}^{d} \sum_{j=1}^{c} \left|w_{ij}^{(t)}\right| + \gamma_2 \sum_{k=1}^{d} \left\|(w^{(t)})^k\right\|_2
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad & Tr(X^T W^{(t+1)} - Y)^T (X^T W^{(t+1)} - Y) \\
& + \gamma_1 \left\|W^{(t+1)}\right\|_1 + \gamma_2 \left\|W^{(t+1)}\right\|_{2,1} \\
& \leq Tr(X^T W^{(t)} - Y)^T (X^T W^{(t)} - Y) \\
& + \gamma_1 \left\|W^{(t)}\right\|_1 + \gamma_2 \left\|W^{(t)}\right\|_{2,1}.
\end{aligned}
$$

The last but one step holds according to Lemma 3.3.1. Therefore, the algorithm decreases the objective value in each iteration. □

In the convergence, $W^{(t)}$, $D_i^{(t)}(1 \leq i \leq c)$ and $\tilde{D}^{(t)}$ will satisfy the Eq. (3.7). As the problem (3.5) is a convex problem, satisfying the Eq. (3.7) indicates that W

is a global optimum solution to the problem (3.5). Therefore, the Algorithm 1 will converge to the global optimum of the problem (3.5). Because we have closed form solution in each iteration, our algorithm converges very fast.

3.3.4 Competitive Feature Selection Methods

To validate the effectiveness of the proposed method on feature selection from aCGH data we compared its performance with existing feature selection methods which have either been effectively used for aCGH feature selection recently or have shown good performance on other related bioinformatics tasks. Specifically, in our experiments we compared with Maximum Influence Feature Selection (MIFS) [30], Relief-F [48], Information Gain (IG) and χ^2-statistic (chi-squared) [49] as implemented in Weka [50] and Minimum Redundancy Maximum Relevance (mRMR) found in [51].

3.4 Datasets

In order to assess the performance of our proposed method for feature selection in aCGH data, we conducted extensive classification experiments where we compared our method with other state-of-the-art feature selection methods that have been recently proposed for aCGH feature selection. For our experiments we used 4 different publicly available aCGH datasets.

Dataset 1: The first dataset contains a total of 75 samples coming from subjects with oral squamous cell carcinoma (SCC) (14 TP53 mutant samples) and healthy subjects (61 wildtype samples). The dataset is available as part of the supplementary material of the publication [34]. Each CGH sample consists of 1979 probes.

Dataset 2: The second dataset has been collected by the authors of [52] to investigate the biological basis between aging and sporadic breast cancer incidence

and prognosis. DNA samples from matched ER+ invasive breast cancers diagnosed in either young (¡45) or old (¿70) women were analyzed with aCGH. The datasets consists of 71 samples, 27 of them coming from young women and 44 from old women.

Dataset 3: Our third dataset, consists of 98 samples of aCGH profiles coming from 3 different types of primary colorectal cancer: metastasis-free, liver and peritoneal metastasis. 36 samples come from patients who developed liver metastasis, 37 come from patients who developed peritoneal metastasis and 25 from patients who remained metastasis-free. The dataset can be found in NCBI GEO database with the code name "GSE20496".

Dataset 4: The forth dataset consists of 101 samples coming from 5 different breast cancer subtypes (basal-like - 23 samples, luminal A - 43 samples, luminal B - 14 samples, ERBB2 - 15 samples, and normal breast-like - 6 samples). Each CGH sample consists of 2149 probes. The dataset can be found in the supplementary data of [25].

3.5 Experiments

To evaluate the performance of our proposed feature selection method we conducted experiments on 4 aCGH datasets, where we used HSR and compared to five other feature selection methods. In our experiments we measured the performance of each of the above methods using SVMs [43] and Logistic Regression (LR) [45] for classification. For the needs of our experiments we used the LIBSVM [53] implementation of SVM with RBF kernel and the implementation of Logistic Regression found in Weka [50]. We evaluated the performance of each of the different feature selection methods on a range of different number of selected features (from 5 to 100). To assess the classification accuracy we performed 10-Fold Cross validation applying each of the feature selection methods on the same data subsets and using the same SVM param-

eters, which have been determined in advance as appropriate for the target dataset, throughout the experiments. Furthermore, to eliminate the effect of randomness, we repeated each 10-Fold CV round 10 times, with different sample distributions every time, and we took the average accuracy. The results are shown in Figure 3.2.

The first dataset contains samples from only 2 different classes (oral squamous cell carcinoma vs. healthy tissue), thus forming a binary classification problem. In this dataset HSR shows superior performance compared to the other feature selection methods for both SVM and Logistic Regression classifiers, especially when using between 30 to 50 features. MIFS and mRMR compete for the second place when using the SVM classifier, while IG, ReliefF and chi-squared have a significantly lower performance. When using Logistic Regression as a classifier MIFS, mRMR, IG and chi-squared show a slightly lower performance than HSR, however there is no clear winner between them. ReliefF shows the lowest performance in this case.

The second dataset is again a binary dataset (breast cancers diagnosed in either young (¡45) or old (¿70) women). In this dataset, when using SVM classifier, HSR and MIFS compete for the first place, whereas the other feature selection methods lag far behind. With Logistic Regression as classifier, the overall performance of all methods is lower at smaller number of features and only when using 65 features and above, HSR shows a clear advantage. As it appears, this is inherently a difficult dataset as there might not be enough biormarkers to differentiate between breast cancers in younger and older women. That leads to a low overall classification accuracy for all feature selection and classification methods.

The third dataset is the first multi-class dataset, containing samples from 3 different types of primary colorectal cancer. In this dataset HSR shows significantly higher performance compared to all other feature selection methods for both SVM and LR classifiers.

Finally, the forth dataset contains samples from 5 different classes, thus forming another multiclass classification problem. In this dataset again HSR feature selection methods show superiors performance for both SVM and Logistic Regression classifiers compared to the other feature selection methods, although as one can see HSR is a clear winner when using Logistic Regression classifier. As for the rest of the feature selection methods, we can see that MIFS, IG and chi-squared show a slightly lower performance compared to the HSR, whereas mRMR and ReliefF perform poorly in this dataset for both classifiers tested.

In total, we see that HSR shows top performance in all different datasets and classification methods used. Especially when we are dealing with multi-class problems, such as in datasets 3 and 4, we see that HSR has a clear advantage compared to existing feature selection methods due to its ability to identify features that may be important for one class but insignificant for the rest of them.

3.5.1 Biomarker analysis

Apart from classifying the tumor tissue samples based on their aCGH analysis, it is of great importance to identify what genomic abnormalities cause the disease itself. In other words we are interested in identifying the biomarkers that may connect certain properties of the genotype with their corresponding effects on the phenotype. Those connections are already known for some disease types. For example, in figure 3.4 we can see the connection between certain genes and diseases as listed in Entrez Genome NCBI Database[1]. The visualizations are made using the on-line Entrez Map Viewer Software[2]. However, for many disease types, their connection to certain

[1]Entrez Genome NCBI Database organizes information on genomes including maps, chromosomes, assemblies, and annotations (http://www.ncbi.nlm.nih.gov/sites/genome).

[2]The Map Viewer provides special browsing capabilities for a subset of organisms in Entrez Genomes. Map Viewer allows you to view and search an organism's complete genome, display

genomic functionalities is yet to be discovered. The BAC/PAC clones used to form the aCGH datasets can help towards this direction. BAC (F-factor-based Bacterial Artificial Chromosome) and PAC (P1-derived Artificial Chromosome) are cloning systems specifically designed at cloning DNA fragments in excess of 100 - 300 kb. In aCGH analysis, BAC/PAC clones are used to measure areas of the genome with increased or decreased DNA copy numbers compared to the normal/control levels. Each clone region can contain one or more genes. Over or under-expression of such genes can lead to cell abnormalities such as tumor genesis. Therefore, CNVs that occur consistently for a certain disease in the genomic area covered by a specific clone can be an indication that the associated genes existing in that area could be related to the disease itself.

Our feature selection method allows us to automatically analyze aCGH data and find clones who's CNVs are related to specific cancer types. The clones are ranked in order of importance based on their predictive power with regard to the examined cancer classes of each dataset. For example, in dataset 3, the clone RP11-47L3 is ranked as the most important with regard to its ability to differentiate between the three different cancer types of the dataset. Increased copy number of the clone shows a strong correlation with colorectal cancer type 1 (liver metastasis), whereas decreased copy number shows strong correlation with colorectal cancer type 2 (peritoneal metastasis). The CNVs of the clone does not show strong correlation with class type 3 (metastasis-free colorectal cancer), (see figure 3.1). The RP11-47L3 comes from locus AC022706 of Homo Sapiens chromosome 17 (see figure 3.5). In the same region lies the gene SLFN5 (schlafen family member 5) which encodes a protein believed to have a role in hematopoeitic cell differentiation. Therefore, this gene and the correspond-

chromosome maps, and zoom into progressively greater levels of detail, down to the sequence data for a region of interest.

Table 3.1. The 20 most important BAC/PAC clones of Dataset 1 and the corresponding genes found in the genomic area covered by each clone.

Dataset 1		
	Clones	**Genes**
1	RP11-43B19	LPAL2
2	RP11-42A17	GABRG1
3	GS1-174H8	BBS9
4	CTB-1O12	FHIT
5	RP11-110I16	RP11-110I16
6	RP11-59E12	LAMA3
7	RP11-52B21	LRCH1, ESD
8	RP11-14I14	JMJD1C
9	RP11-130N6	N/A
10	RP11-283M20	RPS15A, ARL6IP1, SMG1
11	RP11-109D4	RP11-109D4, ARL6IP1, SMG1
12	RP11-119N7	LOC645481
13	RP11-70F16	N/A
14	RP11-221G13	MAMLD1
15	RP11-34J24	VOPP1
16	RP11-162F2	RPS27AP11
17	RP11-88B16	EFCAB5
18	RP11-160L9	CDK2AP2, CABP2, GSTP1, LOC100505621, NDUFV1, LOC390213, NUDT8, TBX10, ACY3
19	RP11-94J8	IL13RA2, LOC100419790, YAP1P2
20	RP11-97P11	LANCL2, VOPP1

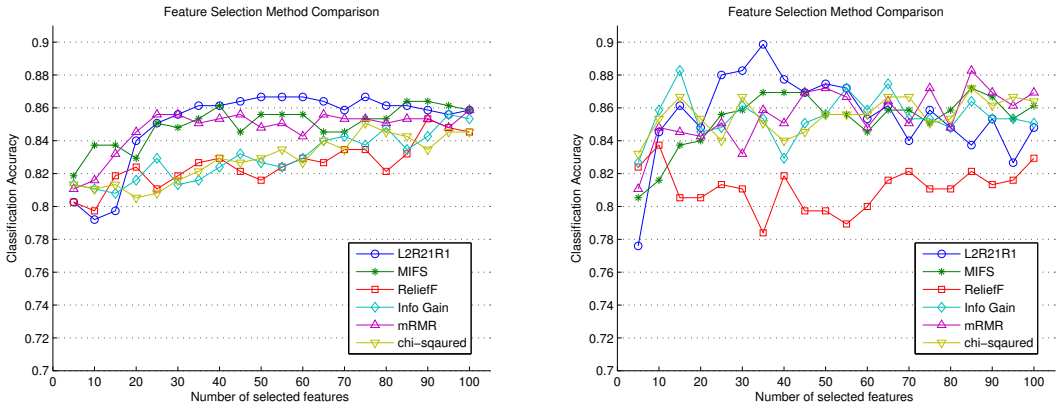
ing encoded protein may be related to the metastasis type developed by the examined patients. Tables 3.1, 3.2, 3.3 and 3.4 list the top 20 clones of each dataset and the corresponding genes found in the genomic area covered by each clone. Where "N/A" appears instead of a gene, it means that there is no known gene in the covered area according to NCBI Genome Entrez Database.

Table 3.2. The 20 most important BAC/PAC clones of Dataset 2 and the corresponding genes found in the genomic area covered by each clone.

Dataset 2		
	Clones	**Genes**
1	RP11-145B20	SLC1A2
2	RP11-568F15	OR10V1, OR10Y1P, OR10V3P, OR10V2P, STX3, FABP5L7, MRPL16, GIF, TCN1
3	RP11-49D19	ZBTB3, POLR2G, TAF6L, TMEM179B, TMEM223, NXF1, STX5, WDR74, RNU2-2, SNHG1, SNORD22, SNORD25-SNORD31, SLC3A2, CHRM1
4	RP11-729B4	MS4A14, MS4A5, MS4A1, MS4A12, MS4A13
5	RP11-77M17	SERPING1, MIR130A, LOC100507106, YPEL4, CLP1, ZDHHC5, MED19, LOC100507231, TMX2, C11orf31, BTBD18
6	RP11-129G17	VN1R55P, RNLS
7	RP11-45L17	C10orf68, ITGB1, LOC100288319
8	RP11-35F11	HRASLS5, LGALS12, TMSL5, RARRES3, HRASLS2
9	RP11-181I11	N/A
10	RP11-61G7	SPAG8, HINT2
11	RP11-40G3	DLG2
12	RP11-48K2	BOD1
13	RP11-206I1	RP11-206I1, LOC100507338, LOC100419850
14	RP11-287G20	CCDC147
15	GS1-54J22	C1GALT1, LOC100505904
16	RP11-39C2	GPR116, GPR110
17	RP11-160A13	PAQR9, LOC100289361, SR140
18	RP11-1L22	GPR39
19	RP11-215H8	ODZ4
20	RP11-39I6	CLTA

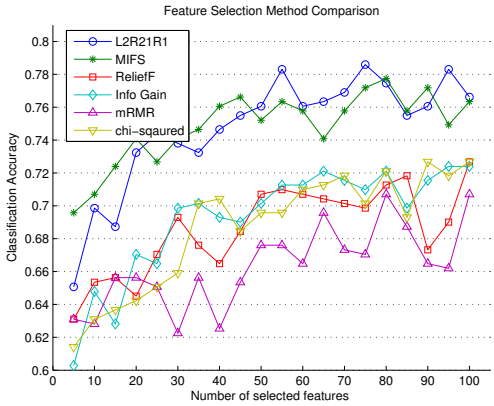
3.6 Discussion

In this chapter we presented a novel feature selection method and compared its performance with existing state-of-the-art feature selection methods. The proposed method, which applies regression based hybrid sparsity regularization to determine the optimal coefficients for the initial set of features, consistently showed superior performance compared to other feature selection methods when user for feature selection in aCGH data. Especially in multi-class problems our method manages to significantly outperform the competitive feature selection methods. Our method is independent of the algorithm to be used during the classification process which makes it ideal for use in combination with different classification methods. Although in this work we examine the performance of our proposed method on aCGH data, it can be also applied in a variety of different data types where feature selection is useful.

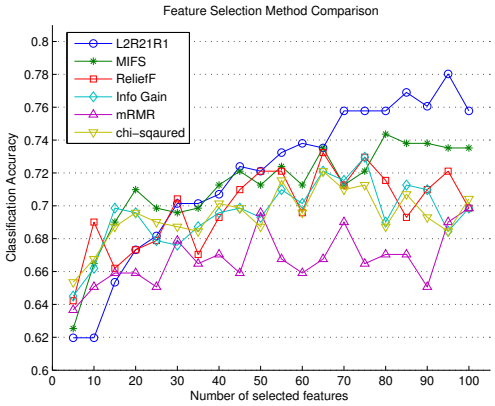


(a)

(b)

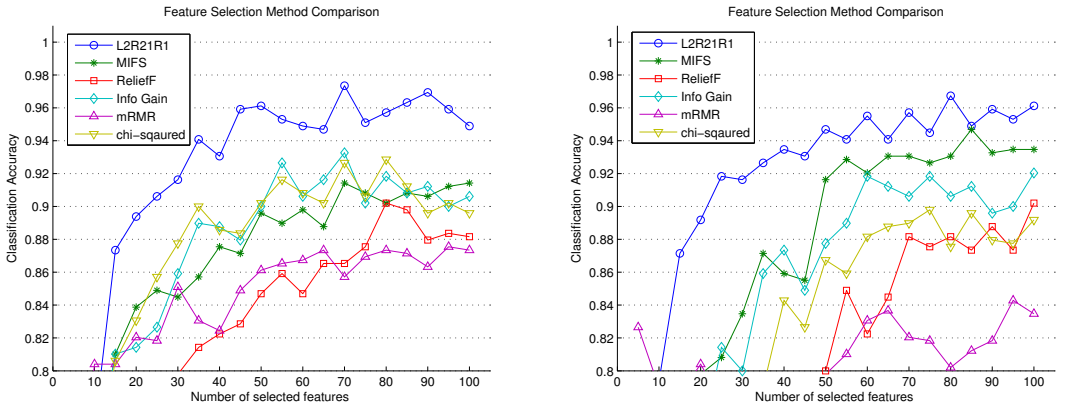


(c)



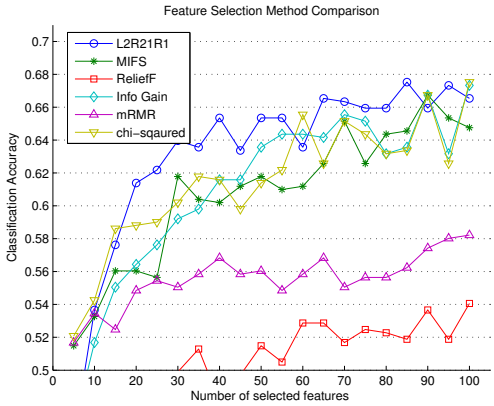
(d)

Figure 3.2. Classification accuracy results for datasets 1 and 2 comparing HSR (L2R21R2) with 6 existing feature selection methods using SVM and Logistic Regression classifiers. (a) Dataset 1 - SVM. (b) Dataset 1 - Logistic Regression. (c) Dataset 2 - SVM. (d) Dataset 2 - Logistic Regression.

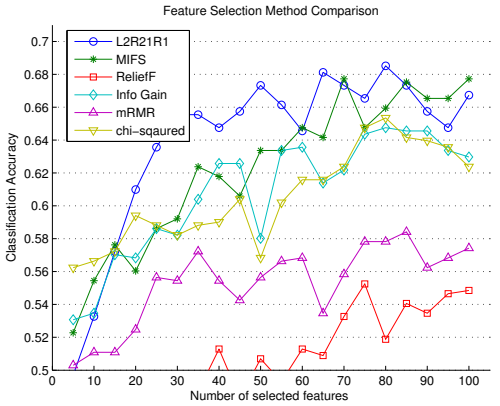


(a)

(b)



(c)



(d)

Figure 3.3. Classification accuracy results for datasets 3 and 4 comparing HSR (L2R21R2) with 6 existing feature selection methods using SVM and Logistic Regression classifiers. (a) Dataset 3 - SVM. (b) Dataset 3 - Logistic Regression. (c) Dataset 4 - SVM. (d) Dataset 4 - Logistic Regression.

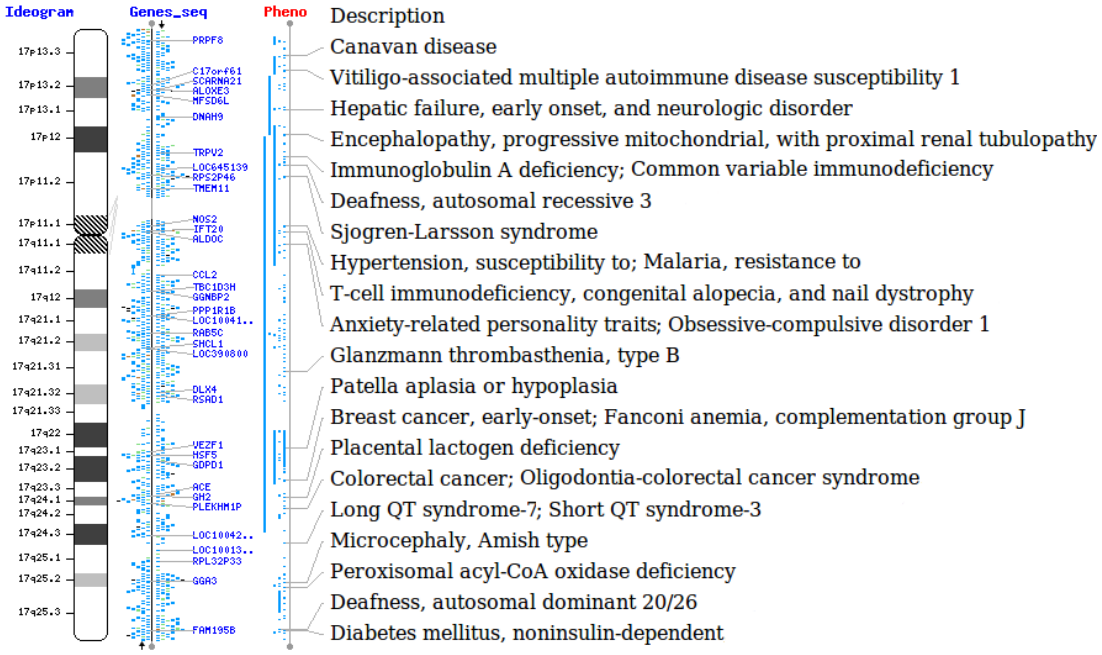


Figure 3.4. Genotype-Phenotype mapping of well known genes and diseases on Chromosome 17, extracted from Entrez Genome NCBI Database.

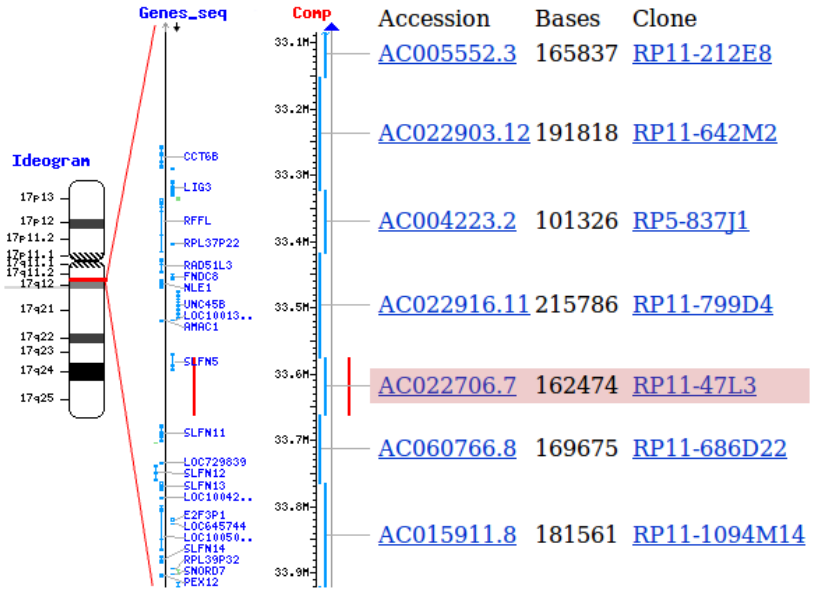


Figure 3.5. Clone-Gene mapping in the region 33,080K-34,650K bp of Chromosome 17. In the genomic area covered by the examined clone (RP11-47L3) we find the gene SLFN5.

Table 3.3. The 20 most important BAC/PAC clones of Dataset 3 and the corresponding genes found in the genomic area covered by each clone.

Dataset 3		
	Clones	**Genes**
1	RP11-47L3	SLFN5
2	RP11-202L1	N/A
3	RP11-213G21	N/A
4	RP11-339F13	EGFR, LOC100507500, LOC100130121, CALM1P2
5	RP11-338H14	N/A
6	CTC-263A14	LOC100131520
7	RP11-359H18	LOC100131479, RPS27P29, VN1R93P, ZNF675, VN1R94P, ZNF681
8	RP11-219A15	LOC266619, LOC353194, LOC400578, LOC147228, LOC339186, CLPSMCR, TBC1D27
9	RP4-552K20	MAGEC3, LOC100420249, MAGEC1
10	RP11-447J13	CADM2, LOC100422711
11	RP11-767J14	N/A
12	RP11-164K24	LOC100506669, LOC283710
13	RP11-125I23	GTF3A, MTIF3
14	RP11-122N14	DMD
15	RP11-326G21	PDE4DIP, LOC100505971
16	RP11-27C22	RP1-27C22
17	RP11-67F24	IL12A, LOC730109, BRD7P2
18	RP11-187L3	CRYL1
19	RP11-426J23	EPHB6, TRPV6, TRPV5, C7orf34
20	RP11-182H20	TTTY8, TTTY7B, TTTY21, TTTY2, TTTY1

Table 3.4. The 20 most important BAC/PAC clones of Dataset 4 and the corresponding genes found in the genomic area covered by each clone.

	Dataset 4	
	Clones	**Genes**
1	RP11-48I18	ZNF423, MRPS21P7, MRPS21P8
2	RP11-58M3	MARVELD3, PHLPP2, SNORA70D, SNORD71
3	CTA-799F10	SHANK3
4	RP11-52K17	RPL5P26, COL13A1
5	RP11-14G23	TDRG1, LRFN2, LOC100505697
6	RP11-105E14	LIX1L, RBM8A, GNRHR2, PEX11B, ITGA10, ANKRD35, PIAS3, NUDT17, POLR3C, RNF115
7	RP11-204D12	PCSK1
8	RP11-44N11	LOC392265, LOC100507001, ZHX2
9	RP11-15L8	LRFN4, PC, RNU7-23P, MIR3163, C11orf86, SYT12
10	RP11-116F9	RPL5P22, PNOC, ZNF395
11	RP11-249H15	CDK18
12	RP11-16A21	LOC100131036, SPIRE1
13	RP11-141N1	LOC100132126
14	CTB-23D20	TAX1BP1, JAZF1
15	RP11-208E21	VPS13B, LETM1P3
16	RP11-33J8	SFMBT2
17	RP11-35I11	N/A
18	RP11-125O21	LOC100131849, KCNS2, STK3
19	RP11-177M14	EYA4
20	RP11-45B19	ZFAT, ZFATAS

CHAPTER 4

ANALYSIS AND FUSION OF HETEROGENEOUS MULTIMODAL DATA

4.1 Introduction

When dealing with problems which involve understanding and classification of human behavioral patterns or medical conditions, in many cases it helps to examine different aspects of the same problem. Such aspects could come from different data modalities collected from the same subjects. The different modalities could be generated by monitoring the subjects with different sensor types simultaneously or by measuring their medical condition indicators with different types of devices or techniques. Often, the different data modalities could be completely heterogeneous and the information that they convey may seem unrelated to each other, however, when suitably combined may significantly facilitate our understanding of the problem or increase the accuracy of the obtained results. In this chapter, we present our work in dealing with two different problems related to human subjects, that is brain tumor typing and detection of sleep abnormalities, and we show how the fusion of heterogeneous data sources can increase classification accuracy.

4.2 Heterogeneous Data Fusion to Type Brain Tumor Biopsies[1]

4.2.1 Problem

Brain tumors are one of the leading causes of death in adults [57]. The potential value of combining high resolution magic angle spinning (HRMAS) proton (1H) Magnetic Resonance Spectroscopy (MRS) and gene expression data for brain

[1]For details about this work see [54, 55, 56].

tumor typing has been previously proposed [58]. Also the molecular classification of brain tumor biopsies using 1H HRMAS MRS and robust classifiers has been recently reported [59]. However, this classification was limited to the binary classification problem of discriminating between tumor types using the one-versus-all classification methodology. Here, we use machine learning algorithms to create a novel framework to perform the heterogeneous data fusion on MRS and gene expression data coming from the same brain tumor biopsies, to identify different profiles of brain tumors. We concentrate on the data fusion for the problem of assigning each sample to one of the multiple possible tumor type classes. Therefore, we select features (biomarkers) from multi-source simultaneously and those selected features are discriminative to all brain tumor types used in our study, not just to individual ones.

4.2.2 Datasets

We used 46 samples of normal (control) and brain tumor biopsies from which we obtained ex vivo HRMAS 1H MRS and gene expression data respectively. The samples came from tissue biopsies taken from 16 different people. Out of the forty-six biopsies that were analyzed, 9 of them were control biopsies from epileptic surgeries and the rest 37 were brain tumor biopsies. The tumor biopsies belonged to 5 different categories: 11 glioblastoma multiforme (GBM); 8 anaplastic astrocytoma (AA); 7 meningioma; 7 schwanoma; and 5 from adenocarcinoma. From the MRS data we extracted and used as features 15 significant metabolites: choline (Cho), phosphocholine (PC), glycerophosphocholine (GPC), phosphoethanolamine (PE), ethanolamine (Etn), γ-aminobutyric acid (GABA), n-acetyl aspartate (NAA), aspartate (Asp), alanine (Ala), polyunsaturated fatty acids (PUFA), glutamine (Gnl), glutamate (Glu), lactate (Lac), taurine (Tau) and lipids (Lip). For the gene expression profiles the original feature space comprised 54,675 genes. We experimented with

feature selection from both dataset types in order to reduce redundancy and noise before using them for classification.

4.2.3 Methods and Experimental Results

4.2.3.1 Stage 1: Experimentation with Existing Feature Selection Methods

At first, because the main goal was to examine the potential gain from combined heterogeneous data for tumor typing, we only performed experiments with the most well studied feature selection and classification methods. The feature selection methods we applied include the filter methods Relief-F (RF) [60], Information Gain (IG) [49] and χ^2-statistic [61], and a Wrapper feature selection method for each classification algorithm. As for the classification methods, we used Nave Bayes [62] and Support Vector Machines (SVMs) [43, 63]. The methodology of our classification framework is summarized in Figure 4.1 and is comprised of 3 main steps: feature selection from each dataset separately, merging of the top features from both datasets, and classification based on the combined feature set ([55, 56]).

We performed experiments to evaluate the classification accuracy when using each of the datasets separately and in combination. For the MRS data we tested our classifiers for the case of using all available metabolites and for the cases of applying each of the 4 feature selection methods. The best accuracy of **78.72%** (Figure 4.2a) was obtained by the SVM classifier by using the wrapper feature selection method. For the gene expression data we followed a hybrid feature selection approach selecting the top 100 genes by using a filter feature selection method and then further reducing the feature number by using wrapper feature selection. The best accuracy we could get for this dataset came from the Naive Bayes classifier and it reached **82.98%** (Figure 4.2b). Finally, we experimented with the combination of the best features

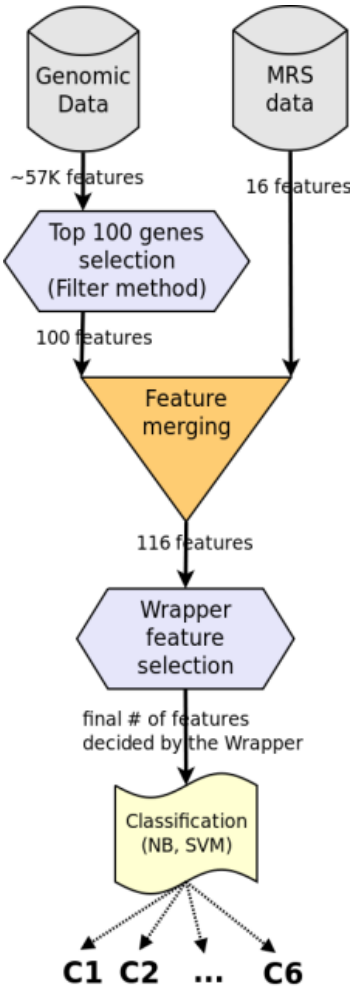
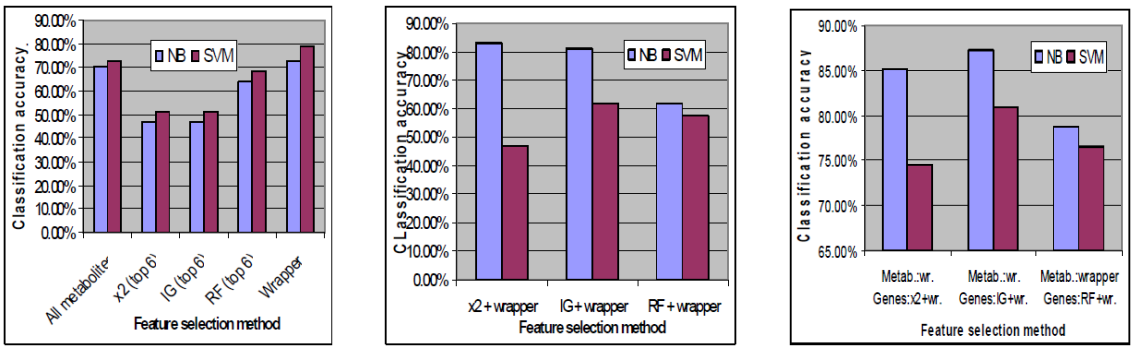


Figure 4.1. Fusion feature selection and classification framework.

drawn from each of the above dataset. In this case both classifiers outperformed the respective best accuracies for the individual datasets. The best result of **87.23%** (Figure 4.2c) was given by the NB classifier when using wrapper feature selection for the MRS dataset and a combination of IG and wrapper feature selection for the gene expression dataset.



(a) (b) (c)

Figure 4.2. Classification results with various combinations of data and feature selection methods. (a) Classification accuracy for the using MRS data only. (b) Classification accuracy using gene expression data only. (c) Classification accuracy using a combination of features from multi-source.

4.2.3.2 Stage 2: Experimentation Sparse Feature Selection

At the next stage we tried to further improve the classification accuracy by enhancing our feature selection methodology. To achieve so we employed a novel sparse filter feature selection method, based on $\ell_{2,1}$-norms minimization, which resulted in a significant increase in the classification accuracy compared to the previous results on the same datasets. This method reduces the feature dimensionality by performing sparsity regularization on the initial feature set which gives a high weight to the most discriminative features and small weight to the rest of them. The optimal weights (coefficients) are obtained by performing $\ell_{2,1}$-norm minimization on the linear regression objective function.

Denote data matrix $X = \{x_1, x_2, \dots x_n\} \in \mathbb{R}^{d \times n}$ and

label matrix $Y = \{y_1, y_2, \dots y_n\}^T \in \mathbb{R}^{n \times c}$. To perform feature selection, we optimize:

$$\min_W J(W) = \sum_{i=1}^{n} \|W^T x_i - y_i\|_2 + \gamma R_4(W) = \|W^T x_i - y_i\|_{2,1} + \gamma \|W\|_{2,1}. \qquad (4.1)$$

Although solving this joint $\ell_{2,1}$-norm problem seems difficult as both terms are non-smooth, it has been shown in [42] that is can be efficiently solved. The other filter methods used were χ^2-statistic (χ^2), Information Gain (IG) and Relief-F (RF) [49].

To compare with the previous classification accuracy results reported on the given datasets [55] we followed the same experimentation process, but this time we replaced the previously used filter feature selection methods with our newly introduced one. Our findings showed, again, that the combination of data from two different sources yields higher accuracy compared to the accuracy that we obtain get by using each of the datasets separately. Also, same as before, our experiments reported perfect accuracy in the ability of the system to differentiate between tumor and non-tumor (normal) samples when the two datasets are combined. For the more difficult task of 6-class classification problem (5 tumor types + 1 normal) though, the use of the new feature selection method significantly increased the accuracy from the 87.23% that was the best previous performance to **95.75%** (Table 4.1). This accuracy was achieved by performing a 10-Fold cross validation on the combined data using Naive Bayes for wrapper feature selection and as classifier to do the final classification. The SVM achieved a relatively lower accuracy due to its inability to be successfully used as a wrapper feature section method because of the high computational complexity required for tuning its parameters.

4.2.4 Biological Meaning

The final set of features that were selected by our system to achieve the above accuracy was a combination of 4 metabolites (Asp, Etn, GPC, PE) and 9 genes (ADM, CD24, ACTB, HSPA1B, CRYAB, MPZ, ABCA2, ID4, PTGDS). The discriminative power of this relatively small set of metabolites and genes may be suggesting that

Table 4.1. Best results for each dataset and each classifier for the 6 class classification task. The feature selection method that achieved the highest accuracy along with the accuracy itself is shown in each table cell.

Classifier / Dataset	**NB**	**SVM**
Mebolites only	Wrapper 72.34%	Wrapper 78.72%
Genes only	χ^2 + Wrapper 82.98%	$\ell_{2,1}$ + Wrapper 68.09%
Combined	$\ell_{2,1}$ + Wrapper **95.75%**	IG + Wrapper 80.85%

they can be used as possible Biomarkers related to the development of brain tumors and further investigation of their properties would be worthwhile.

4.3 Non-Invasive Analysis of Sleep Patterns via Multimodal Sensor Input

4.3.1 Introduction

According to the American Academy of Sleep Medicine, there are 81 official sleep disorders, presented in [64]. 70 million people in the USA have a sleep disorder, the vast majority of which remain undiagnosed and untreated. It is estimated that sleep related problems incur $15.9 billion to national health care budget. There is then great need for automatic non-intrusive methods for sleep disorder recognition, that patients can use in their homes. This would not only help decrease health care costs but also increase the number of diagnosed patients.

Another reason why sleep disorder detection is important is the fact that it is related to other potentially more serious medical conditions. According to [65], results of their study involving 1506 participants (out of which 83% reported some medical condition) show that sleep disorders are related to comorbidities rather than age. This is most likely because major comorbidities such as stroke, heart disease, osteoporosis

or arthritis impact the patients' quality of sleep. Detection of sleep disorders could therefore be an indication of another important disorder.

[66] studied 917 patients from a wide range of ages and suggest that patients with chronic sleep disorders are more likely to have depression and in fact about 1 in 4 patients who went to a sleep disorder clinic admitted to be experiencing depression, although only 3.5% were found with moderate to severe depression.

We propose a non invasive system that is able to analyze and recognize sleep patterns which can be further utilized to detect various types of sleep disorders. The first sensor that we employ is a bed pressure mat (product of Vista Medical Ltd[2]) where the patient sleeps. The second sensor is the Kinect 3D image acqusition device by Microsoft [9]. Our aproach is strongly motivated by the fact that by combining the information acquired by the two sensors it is possible to attain better results than from a single one, due to the complementarity of the acquired information. Indeed, the pressure mattress is very reliable in capturing the information about the users' body parts that are in contact with it, but cannot provide any information about the rest of the body. On the other hand, the Kinect cannot see the body parts touching the mattress, but can provide rich data about the rest of the body parts that are visible. To the best of our knowledge this is the first such multimodal approach for non-invasive recognition of sleep patterns.

We analyzed the acquired data using Supervised Machine Learning techniques and the system classified the sleep patterns of the user in one or more predefined categories regarding both *posture* and *motion*. In this work we experimented with data collected from seven individuals. The different patterns included periods of normal sleep and periods of abnormal sleep such as restlessness, and frequent changes

[2]http://www.pressuremapping.com/

of body position. Preliminary results show that our system is able to successfully recognize sleep patterns and classify them among a predefined set of categories.

4.3.2 Related Work

Related research has focused on detecting various parameters of sleep for humans and animals as well as sleep quality and body posture recognition. More specifically, studies on rodents focus mainly on detecting if the animal is asleep or awake using piezoelectric films, used as a filtering stage for traditional classifiers using Electroengephalograms (EEG) and Electromyograms (EMG) [67]. The authors use EEG signals, preprocessed using Fast Fourier Transform (FFT), Principal Components Analysis (PCA) for feature selection and classified using the k-Nearest Neighbour (k-NN) algorithm. [68] also uses EEG and other signals and Markov modeling techniques to classify normal and abnormal human sleeping patterns. These types of signals require traditional Digital Signal Processing techniques such as Discrete Fourier Transform (DFT) and PCA for extracting meaningful features and k-NN or Artificial Neural Networks for the recognition step. Nevertheless, these methods require sensors or cables attached to the skin of the subject which is not acceptable for assistive pervasive applications. Other researchers use additional types of data, such as oxymetry information to detect respiratory abnormalities [69]. The authors evaluate classification results using spectral and nonlinear analysis for feature extraction and Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), k-NN and Linear Regression (LR) for classification. In [70] the authors try to assess sleep quality using near-infrared video only. The authors apply a homomorphic filtering technique to tackle the problem of over exposure in the center, common in near-infrared cameras. The authors also learn a threshold to differentiate noise from actual motion, since this type of cameras have ver low signal to noise (SNR) ratio.

They then use the Motion History Image (MHI) technique that provides direction of movement to identify motion.

Pressure has also been used to infer if the subject is asleep or awake by detecting movements and respiration of rodents. There exists one previous approach to our knowledge that recognizes sleeping posture of humans using pressure sensors. More specifically 32 pressure sensors where used to record the pressure pattern of the subject at a particular pose and Naive Bayes as well as Random Forests where used for classification and compared to each other [71]. In [72] the authors use a pressure mat to identify sleeping postures of babies possibly assisting prevention of Sudden Infant Death Syndrome. The authors collected the data from a one year old baby freely moving on the pressure mat and after a feature selection stage they classified each posture using majority vote of k-NN, SVM, linear and quadratic classifiers and then applied a sliding window algorithm to eliminate possible mis-classifications.

In our literature survey we didn't find any other non-invasive method that would be able to combine the benefits of a contact-based sensor such as a pressure mattress with the merits of non-contact sensors such as 2D or 3D cameras. Furthermore, the related work is rather limited to posture identification and does not cover motion patterns, which may be of importance. In this work we aim to cover this gap.

4.3.3 Multimodal Sleep Pattern Analysis

4.3.3.1 Description of Datasets

For the needs of our experiments we collected data from 7 different individuals simulating their sleep habits. Each individual lied on the bed for a period of time and performed the actions that they would normally perform if they went to bed. That involved getting in bed, staying still for periods of time in different postures, changing

body postures, moving parts of the body like the arms or the legs and getting out of the bed. The different actions performed during that period of time were recorded using 2 different sensors. The first one was a bed pressure mat (see section 2.2.1.1) that we put under the sheets, and the second one was a Microsoft Kinect sensor (see section 2.2.1.2) that we mounted on the ceiling. The recorded data were then manually annotated according to the various classes of interest, such body posture, motion occurrence, etc. More details about the collected data and the methods used to collect them can be found in section 2.2.1.

4.3.4 Data Analysis and Classification

The detection/recognition of sleep disorders usually boils down to the recognition of a set of symptoms that are related to a specific sleep problem. Such symptoms are: how long it takes for the person to fall asleep, how many times (if any) they wake up during the night, how often do they move during their sleep time, how many hours on average do they sleep, etc. These indicators are difficult to monitor at home. Our immediate goal is to create a system that can recognize these indicators and make them easily accessible to the physicians. The long term goal is to create a system that will be able to automatically detect specific sleep disorders based on training data from previous known cases.

To achieve our goal we break our problem into a set of classes and we employ a combination of rule based and supervised learning methods to classify the various instances into one of those classes. To evaluate the classification accuracy, we perform leave-one-out cross validation experiments where every time we test the classification accuracy on the data collected from one user, by training it on data collected from the other users.

In more detail, we are attempting to recognize the following situations: (i) if the person is in bed or not, (ii) when does motion occur while in bed, (iii) what type of motion is that, and (iv) while the person is not moving what is their body posture in bed. Being able to detect and recognize the above situations and then combining them together can be a very rich information source with regard to the symptoms that we want to identify. In the following sub-sections we will describe how we approach each of the above situations and how efficient our system is in terms of recognition accuracy.

4.3.4.1 Detecting if the person is in bed or not

The first case of interest in our experiments would be to detect when the person is in bed or not. This is useful in cases, for example, where we want to know how many hours in total does the person spend in bed and how often do they get up during their sleep time. It turns our that this is a very easy problem to solve by just using the bed pressure mat. All we had to do is just define a threshold of the total amount of pressure that we get in the pressure mat. If the total pressure exceeds that threshold it means that the person is in bed. Using this approach we got 100% accuracy in detecting if the person is in bed or not in our experiments. Note that we did not consider cases where somebody puts something heavy on the bed that might confuse our system, since we assume that participants are willing to be examined and they are not willing to mislead the system.

4.3.4.2 Motion Detection

Another case of interest, is to detect when motion occurs while the person lies in bed. The detection of motion can be related to various sleep disorder symptoms.

For example, it can be an indication of how long does the person take to fall asleep after they go to bed, or how often do they wake up during the hight.

To detect motion we used the standard computer vision technique of frame differencing. That means that we compared consecutive frames by subtracting the frame n from the frame $n+i$, where $i \geq 1$ depending on the frame rate, and summing up the absolute differences. The value of that sum S is a very good indicator of the existence of motion in the time slot between the two frames.

For example, by using the only bed pressure mat, this can be achieved by calculating the sum of absolute differences of the values of each of the 1024 pressure sensors between consecutive frames represented as vectors. Assuming a frame vector $X_k = \{x_1, x_2, \ldots, x_n\}$, where $n = 1 \ldots 1024$, at each time point k, this sum S can be can be calculated as follows:

$$S = \sum_{i=1}^{n} |x_{k+1,i} - x_{k,i}| \tag{4.2}$$

It turns out that motion can be easily detected by specifying a threshold T on the value of S. If S becomes greater than T, the subject is moving. The optimal value of T can be calculated from the training dataset and it is almost constant among subjects of similar weights. Figure 4.3 shows a graph of the values of S over a period of about 1500 frames obtained from one of the subjects. The green horizontal line defines the threshold.

Exactly the same approach can be used on the data collected from the Kinect sensor. The only differences compared to the pressure mat frames, are the frame rate, the frame resolution and the pixel value range. However, the formulas to calculate the sums S and the optimal threshold are exactly the same.

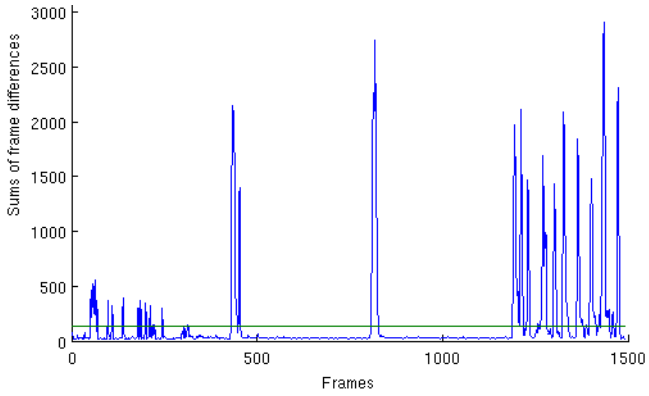


Figure 4.3. Detection of motion using the sum of absolute frame differences (S) and a threshold $T = 130$.

Using this approach we classified each frame in the stream as containing motion or not. We tested our system's accuracy against the manually annotated data where a human had specified the time points where motion occurred. We experimented using the pressure mat only, and the combination of pressure mat and Kinect. To combine the two different data sources we re-samples the Kinect data to meet the pressure mat frame rate (3 Hz) and then we aligned the frames using their timestamps. Each frame was classified as containing motion, if the value of S in either of the two data sources exceeded the predefined threshold.

Using the pressure data only, we achieved an average motion detection accuracy of 96.83%, whereas adding the Kinect data the accuracy increased to 97.57%. The increase in accuracy can be attributed to cases where a motion (e.g. hand movement) is not strong enough to be detected by the pressure mat but it can still be detected by Kinect. The majority of the misclassified frames were spotted either at the beginning or at the end of movement of the subject where the levels of motion are very low. Hence, some of those might have actually been miss-annotated during the manual annotation process. In any case, the results of motion detection accuracy can be considered satisfactory.

4.3.4.3 Recognition of motion types and body postures

After detecting motion, our next step was to recognize the motion type, when motion occurred, and the the subject's body posture when there was not motion. To do that, we first used our motion detection method to segment the data steams into sequences of frames which are part of a motion and sequences of frames where there is no motion. Then we classified each of those sequences into one of the motion classes or body posture classes.

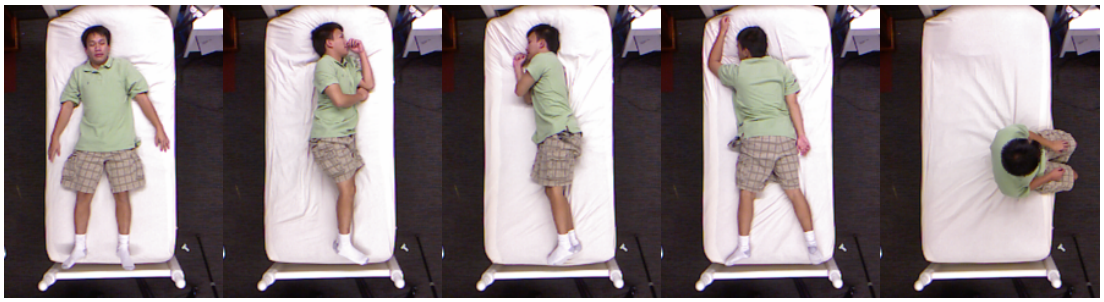
The basic motion classes that we defined were the following:

1. Changing body posture.
2. Moving arms or Legs.
3. Getting in bed or out of bed.
4. Making bed.

The first class refers to the case where the subject is changing sides, for example, they are sleeping on their back and then they turn their left. The second class refers to more subtle motion types where the subject moves a part of their body, usually a limb, but they don't completely change their body position. The third class, occurs when the bed gets in or out of the bed. This motion type differs from the previous two considerably. The last class, refers to the case where the person is not actually in bed but there is still some type of motion detected by the pressure mat or the Kinect. This is usually the case when someone makes their bed.

Regarding the body postures we defined the following classes:

1. Back
2. Left Side
3. Right Side



(a) (b) (c) (d) (e)

Figure 4.4. The 5 different body postures. (a) Back. (b) Left Side. (c) Right Side. (d) Stomach. (e) Sitting.

4. Stomach

5. Sitting on bed

The first four classes cover the basic usual sleeping postures, whereas the fifth class occurs when the subject is on the bed but they are not actually lying on it. Such cases usually occur when the subject is about to get in or out of the bed, but there could also be cases where they don't feel good and the temporarily get up for a few seconds. Figure 4.4 gives an overview of these 5 postures.

To recognize the body postures, we experimented with two different techniques. The first one is a Computer Vision based technique, called Template Matching (TM), which has been used in face detection [73] and other similar applications. The idea behind this technique is that for each posture you pick a representative frame to use it as a template, after possibly cropping it appropriately, and then for every other frame to be classified, you compare it with all the templates and see which one matches better according to some distance criterion. In our case, we used the simple frame difference as a distance criterion. That means we calculated the sum of absolute differences of each pixel of the template subtracted from the corresponding

pixel in the frame to be classified. To accommodate for cases where the subject lied in a different position of the bed compared to the template or they were taller/shorter compared to the subject used in the template, we tried different scales and different centering position of the template.

The second technique that we used was based on supervised learning. In order to perform supervised learning, we converted each frame to a a feature vector where each pixel represented a feature. To remove redundant features and reduce noise before classification we performed a Principal Components Analysis (PCA) [74] transformation on the data. In addition, we calculated the Central Image Moments of the original frames and we added those as features to the feature vector the resulted from PCA. An image moment is a certain particular weighted average (moment) of the image pixels' intensities. The advantage of Central Moments is that they are translation invariant which could be useful in cases where the subject is lying in an unusual position of the bed. For a digital grayscale image with pixel intensities $I(x, y)$, the raw image moments M_{ij} are calculated by

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{4.3}$$

The central moments can be calculated using the following equation:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{4.4}$$

where $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the components of the centroid. We used Central Moments of order up to 2, which yields 8 different moments. For the data coming from the Kinect we assumed a near-constant background and we defined the region of interest to be the area that covers the dimensions of the bed.

In our experiments we evaluated our methods using each of the data sources separately and in combination. To combine features from the two different data sources we aligned the pressure and the depth sensing frames using their timestamps and we created a composite feature vector which included the top Principal Components and the Central Moments of each pair of frames. To deal with the difference in the frame rate, we re-sampled the depth sensing stream to reduce its frame rate. The classification methods that we used to classify each single frame were based on the well known KNN [75] (we used 10 neighbors) and Linear Kernel SVM [43] algorithms.

In order to recognize the body posture in the sequence of frames, in both the cases of template matching and supervised learning, we first classified each of the frames in the sequence to one of the predefined classes and then we used majority voting to decide the final posture class.

At each round we trained our system using data coming from 6 our the 7 users and classified the motion sequences of the 7th user. This ensures that if the system is to be used in real life, it can be trained off-line in advance and it does not require any re-training for the specific user.

For the classification of the sequences of motion frames into one of the 4 classes we used Hidden Markov Model (HMM). A HMM is a statistical model of a system having hidden states and operating under the Markovian assumption. HMMs have been proven to model effectively temporal sequences as well as other forms of sequential data. The models are trained using the Baum-Welch algorithm that calculates their parameters. As for the recognition step, it is done using the Forward-Backward algorithm [76].

For the KNN classifier, we found that the combination of the top 40 principal components from each data source plus the central moments for each frame gave us the best classification accuracy. Similarly for SVM we used the top 30 principal com-

ponents plus the central image moments. For the classification of motion using HMM we used the top 7 principal components plus the central moments from the pressure sensing datasets and the top 14 principal components plus the central moments from the depth sensing datasets.

Table 4.2 presents the classification accuracy results for each user and the weighted average accuracy, where the weight represents the number of instances per dataset. In the different columns of the table we present the result for body posture recognition and motion type recognition, separated by the classification algorithm that was used and also by the type of data source that was used to perform the training and testing. At each experiment we evaluated our system using the pressure sensing (P) data only, the depth sensing (D) data only and their combination (C).

As one can notice, combining the two different data sources by fusing their features gives the best classification accuracy in most cases. Also, with the exception of Template Matching (TM), using the pressure sensing data alone to recognize body postures and motion types gives better accuracy compared to using the depth sensing data alone. The supervised learning methods (KNN and SVM) outperform the template matching classification in the majority of the cases. In our experiments we constructed the templates from one user (User 1) and we applied those same templates to all the other users. That is the reason why template matching works particularly well on "User 1". Since, the template construction only requires the capturing of one frame for each posture and some cropping to match the body dimension, it would not be unreasonable to construct new templates for each new user in real life.

4.3.5 Discussion

In this section we presented our work on analysis of sleep patterns using non-invasive sensors and applying a combination a rule based and machine learning meth-

Table 4.2. Classification accuracy results for Body Posture and Motion Type recognition. "P" as a column title denotes that only pressure sensing date were used, "D" denotes that only depth sensing data were used and "C" denotes that a combination of pressure and depth sensing data were used. For the posture recognition, the best accuracy per data source is in boldface and the best accuracy across the different classification methods is underlined. For the motion recognition, the best accuracy per data source is in boldface.

	Posture Recognition									Motion Recognition		
	TM			KNN			SVM			HMM		
	P	D	C	P	D	C	P	D	C	P	D	C
User 1	87.75	**91.83**	89.79	**83.67**	57.14	**83.67**	89.79	87.75	**91.83**	80.39	74.51	**92.15**
User 2	47.72	56.81	**77.27**	**90.90**	77.27	88.63	81.81	77.27	**84.09**	95.74	76.59	**97.87**
User 3	31.91	57.44	**65.95**	95.74	**97.87**	95.74	91.48	89.36	**93.61**	94.23	78.84	**96.15**
User 4	52.17	63.04	**84.78**	**89.13**	67.39	86.95	89.13	91.30	**93.47**	75.51	63.26	**79.59**
User 5	30.43	08.69	**47.82**	69.56	**73.91**	69.56	78.26	56.52	**86.95**	90.90	27.27	**95.45**
User 6	30.76	33.33	**66.66**	**56.41**	41.02	53.84	51.28	**64.10**	56.41	76.08	65.21	**78.26**
User 7	57.14	52.38	**73.81**	73.81	**92.85**	76.19	90.47	76.19	**92.85**	95.45	54.54	**86.36**
Average	53.10	64.82	**83.79**	**81.38**	72.76	80.69	82.76	79.66	**86.21**	86.50	66.24	**89.07**

ods. Our experimental results on real user datasets show that the task of analyzing sleep patterns with the intent to detect symptoms related to sleep disorders can be successfully achieved. Although the available dataset was relatively small, the classification accuracy results are promising and show that the proposed tools and methods could be used in the future for the detection of sleep disorders and other related diseases affecting sleep quality. To this end, further experimentation with bigger datasets, extended recognition categories and improved fusion methodology would be of high interest.

CHAPTER 5

DISCUSSION OF FRAMEWORK AND EVALUATION

5.1 Introduction

The computational framework proposed in this work, aims at creating the infrastructure for effective human-centered data analysis in order to provide enhanced assistive services to humans. Our main focus falls on how to collect and analyze data coming from monitoring the human behavior inside an assistive environment as well as data coming from their medical condition. The ultimate goal is to create an environment where pervasive technologies will seamlessly provide services to humans in an automated and non-intrusive way. In order for such an environment to be successful in practice, understanding human's behavior and their medical condition is not enough. There are many other parameters that are as important in determining the final success of the system and its adoption by the end users. In this chapter we make an attempt to look at the bigger picture and identify those parameters.

We use as a workbench the setting of an assistive environment and we approach the overall existence of a human-centered computational framework from the software engineering point of view which takes into consideration not only the computational aspects of such a system but also factors such as security/privacy and cost to build and maintain such a system. Finally, in order to ensure the quality of the services provided and the viability of such a system we suggest a set of metrics that need to be used by its creators both during the building process and during the final evaluation of the system.

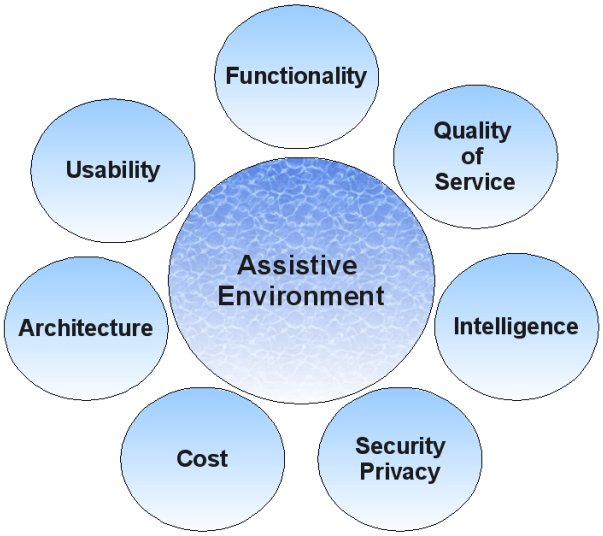


Figure 5.1. The basic attributes of the framework.

5.2 Evaluation of a Computational Framework for Assistive Environments

An assistive environment can be successful only if its potential users are willing to adopt it. This section identifies a set of attributes that are considered critical to user adoption. Sample metrics, as well as possible approaches to measure them, are suggested to quantify those attributes. In the following, we divide these attributes into the following seven categories, namely, functionality, usability, security and privacy, architecture, intelligence, quality of service and cost, and discuss each of them in details. Figure 5.1 gives a visual overview of these attributes.

5.2.1 Functionality

A computational framework targeted to assistive environments must perform correctly in order to serve its purpose, i.e., facilitating the patient's independent living. More importantly, failure in an assistive environment could carry severe consequences. For example, if an acute event is detected by the chest belt, an emergency signal must be sent to a base-station, which should further generate an alarm to alert the caregiver

and if needed, the staff in an emergency room. If the acute event is not detected, or if the emergency signal is not sent timely, the life of the patient may be in danger. Therefore, the evaluation of whether an assistive environment can perform its tasks correctly is at the very core of the evaluation of an assistive environment.

The proposed framework identifies the following major attributes to be used for functional evaluation.

- *Correctness*: A task is implemented correctly if it delivers the required functionality as specified in the requirement document. Ideally, this attribute can be measured by the ratio of the number of tasks that deliver the expected results over the total number of tasks that can be performed in an assistive environment. In practice, the total number of tasks is often difficult to derive. One possible approach is to generate a set of test scenarios that exercise a representative set of the tasks, e.g. based on the functional requirements, and then check how many of those scenarios can be performed correctly. In addition, feature-specific metrics can be developed. An important feature is that the event recognition component must be able to correctly identify events that occur in the environment, based on the activities being monitored. A possible metric for this feature is the ratio of the events that are recognized correctly over the total number of events that occur in the environment.
- *Robustness*: This attribute refers to the ability of an assistive environment to deal with unusual situations [77]. In particular, can faults that may occur or exist in the environment be tolerated? There are two major types of faults to consider: (1) User errors, i.e. mistakes that a user may make when performing a task, e.g., a user may have pressed a button that is not supposed to be pressed given his or her situation. Considering that the users are typically not familiar with technology, user errors are particularly common in assistive environments.

On the one hand, assistive environments should be designed in a way such that user errors are prevented from happening in the first place as much as possible. On the other hand, the system should be able to continue to operate correctly even in the presence of a user error. (2) Device failures. An assistive environment often consists of many small devices that may be subject to failures due to either malfunctions or adverse conditions in the environment. In the case of the assistive apartment environment, a sensor in the data collection component may give an incorrect reading due to some environmental noise or may have gone down due to the depletion of its battery. The failure of one or a few devices should be tolerated, or its impact should be limited as much as possible, in an assistive environment.

Robustness can be difficult to measure precisely because, for example, there can be an infinite number of ways for a user to make mistakes. One possible approach is to generate a set of test scenarios to exercise failures that have a high probability to occur based on an operational profile or based on a careful analysis of the vulnerability of the devices deployed in an assistive environment. The percentage of those test scenarios that can be tolerated by an environment can be used as a possible indicator of the robustness of the environment.

- *Reliability*: This attribute refers to the sustainability of an assistive environment. That is, how long can the environment operate continuously without breaking down? Many assistive environments are designed to monitor the patients' daily living continuously, where a reset can be very inconvenient. In addition, as discussed earlier, assistive environments can be safety-critical, and unexpected breakdowns may have severe consequences. One possible metric for reliability is mean-time-to-failure, i.e., the average time a system can operate continuously before a failure occurs. The key to measure mean-time-to-failure in

an assistive environment is to build an operational profile that is representative of the way in which the environment is used in real life.

5.2.2 Usability

Usability is one of the most important concerns in assistive environments. There are two major reasons. First, assistive environments target a special group of users who are typically not familiar with technology, and may even have mental and/or physical challenges to learn and memorize instructions [78]. Second, the purpose of assistive environments is to assist, rather than create new challenges, in one's daily life. This purpose would be easily defeated if an assistive environment were difficult to use. A key to achieving usability is to make the technology invisible. That is, tasks should be performed in a natural way, i.e., with minimal deviation from how an average person would expect these tasks to be performed by intuition [79]. Related to the above is the fact that an assistive environment should be easy to use for both the patients and the caregivers.

The proposed framework identifies the following major attributes to be used for usability evaluation:

- *Ease of Use*: This attribute consists of several aspects. The user interface of an assistive environment should be easy to navigate. In particular, a user should be able to quickly find commonly used operations. If a sequence of operations needs to be performed to accomplish a given task, then the order in which those operations are performed should be made as straightforward as possible, and the sequence should contain as few operations as possible. If certain input can be derived from context, then it should be done so, instead of asking the user to provide it explicitly. Hints and help should be made readily available, especially for less straightforward operations.

Note that an assistive environment should be easy to use not only for the patient who lives inside the environment, but also for the operators who help to set up and maintain the environment. That is, ease of use implies easy to set system up, easy to maintain, easy to update and easy to learn how to use it. One possible metric for ease of use is the length of the learning curve for a typical user. That is, how long does it take the user to learn the use of an assistive environment? Metrics like the average length of navigation, the average number of steps required to perform a common task, can also be used. However, as this attribute largely deals with user perception, a completely objective measurement would not be possible. Having a group of testers who is representative of the target user base is the key to mitigate the potential variations in user perception.

- *Accessibility*: Assistive environments target a special user group in which many people have mental and/or physical challenges. The user interface of an assistive environment should be made accessible to accommodate the special needs of those people. For example, if the user has difficulties to read the screen, then an audio-based interface may be employed to better interact with the user. Some metrics that can be used to measure accessibility include the number of available accessible options, the number of transformations that are available between different options, and the degree of transparency between those transformations.
- *Non-obtrusiveness*: To maximize the utility of an assistive environment, it is often necessary to be proactive. For example, it is desirable to remind a person who suffers Alzheimer's disease of taking medicine at a regular interval. However, there is a fine line between being proactive and obtrusive. People tend to reject systems that they consider to be obtrusive [80].

This attribute depends on user perception to a large degree, and is thus difficult to measure on a purely objective basis. In particular, the same operation might be considered obtrusive by some people but not by other people. One possible approach to measure obtrusiveness is to identify a group of testers who are representative of the user base and then collect feedback from them.

5.2.3 Security and Privacy

Security and privacy attract more attention when a system involves remote users and when data are shared with other institutions, even for the research purpose. Secure communication, data access control, and robustness against certain attacks are among the most important aspects to be evaluated.

The proposed framework identifies the following major attributes to be used for security and privacy evaluation:

- *Violation reports*: The number of security violation reports (or breaches) and the number of privacy violations could be used to measure the accomplished strength of security and privacy protection.
- *Configurable privacy/access control*: Users can customize policy agreements to grant access and release data; they configure setting files to choose what types of data are sharable with his physicians and other researchers and what types of access they can have.
- *Encryption strength*: Robustness of security & privacy control against crypt-analysis depends on the encryption strength. The length of common module for public/private key pairs can be used for measuring the strength. The password setting could be measured by weak, fair, and strong according to the combination of characters against off-line dictionary attack.

5.2.4 Architecture

Architecture refers to the interconnection of the major components in an assistive environment [81]. An assistive environment often consists of a number of hardware components, e.g. various types of sensors, which are heterogeneous in nature. This calls for an open architecture that allows those components to work together in a seamless manner and in a way that can be easily configured and extended.

The proposed framework identifies the following attributes to be used for architecture evaluation:

- *Modularity*: Modularity is one of the most fundamental principles underlying modern system designs [82]. The idea is to make each component a relatively independent module by reducing the coupling between different components. Doing so makes it easier to change or replace individual modules with minimal effect on the rest of the system. For example, in the assistive apartment environment in Section 3, a modular architecture would allow a data mining component to be easily replaced by another one that employs a different algorithm, or a different type of sensor to be added into the environment. Modularity can be measured by the average number of other modules with which each modular has a direct or indirect dependency relation. The dependency relation between modules can be derived either by analyzing the source code, if available, or by conducting experiments.
- *Interoperability*: An assistive environment may interoperate with other systems. For example, in the assistive apartment environment in Section 3, the base station needs to interact with the server in the emergency room. In addition, within an assistive environment, different components need to interact with each other, and those components may come from different vendors. For example, in the assistive apartment example, the data collection component needs to

work with different types of sensors. The key to achieve interoperability is to define a standard interface (or protocol) so that different parties can speak the same language. Interoperability can be measured by the number of interfaces that conform to a standard. One way to check if an interface of an assistive environment conforms to a standard is to perform conformance testing, i.e., having the environment actually work, at the interface point, with a third-party component or system that is known to be conforming to the standard.

Note that modularity and interoperability are orthogonal attributes, in terms that the former characterizes an assistive environment from a static perspective while the latter does so from a dynamic perspective.

5.2.5 Intelligence

Since assistive environments are a special case of so-called "smart" environments they heavily rely on techniques for inference and automated decision making. Those techniques are based on other well-studied areas such as Machine Learning, Pattern Recognition, Data Mining, and Information Retrieval. These areas of study are known to produce output which is not deterministic and is greatly affected by the type and amount of data to be processed, the noise that is introduced during data collection, the system training procedures and other similar factors that cannot be fully controlled. As a consequence the decision making process does not always produce optimal results and actually in some cases the decisions made may completely contradict with the common logic. However, such techniques are inherent in any smart environment and therefore we need to find ways to assess their efficiency and acceptability by the end users.

Because intelligence is a highly subjective concept and is related to high level cognitive procedures, one way to judge the "artificial intelligence" of a system is to

get feedback from its users. Therefore, we suggest that there should be an intuitive and user friendly mechanism to receive feedback from the users in an assistive environment. We do not expect the users to give feedback for every single decision of the system, but only for the negative cases. For the cases that the users did not give a negative feedback we assume that the decision/action of the system was correct. Note that the user who provides the feedback can be the patient who is being monitored, or a healthcare professional who might be remotely monitoring the patient using the assistive environment.

By quantifying the user input we can assess the intelligence of the system using well known statistical metrics for the evaluation of the system's decision making and predictive performance. Such metrics can include Accuracy, Precision, Recall, Sensitivity and Specificity [83]. For example, one of the functionalities of the assistive apartment environment described in section 2 is to detect dangerous situations for patients with memory problems and fire an alarm. Such dangerous situations can be scenarios like: "the patient is lying on his bed and the stove is on for more than 45 minutes". This could mean that the patient has gone to sleep and has forgotten the stove on. In order to be able to tell if that is the case the system needs make a decision based on behavioral patterns of the patient. In case of a false alarm, the user that will turn the alarm off should also have an option to notify the system that this was a false alarm. In that case the decision made by the system will be logged as false positive. If we define a case of emergency as a case where the system needs to detect it and report it as an example of positive value, then we can measure the system's intelligence as follows:

True Positive (TP): Emergencies successfully reported.

False Positive (FP): No real emergencies reported as emergences.

True Negative (TN): No emergency cases not reported.

False Negative (FN): Emergencies not reported.

A set of metrics than can be used to measure the intelligence of the system are the following:

- *Accuracy*: $\frac{TP+TN}{TP+FP+TN+FN}$
- *Precision*: $\frac{TP}{TP+FP}$
- *Recall* or *Sensitivity*: $\frac{TP}{TP+FN}$
- *Specificity*: $\frac{TN}{FP+TN}$

5.2.6 Quality of Service (QoS)

In general terms, QoS can be defined as an agreement for service between a customer and a provider in which there is a guaranteed level of performance. QoS has been studied extensively in application domains such as Networks and Multimedia [84, 85]. Key aspects have been identified [86] to enable QoS in ubiquitous and heterogeneous environments, such as QoS specification, translation, setup, and adaptation; and specific solutions have been proposed [87]. However, many design considerations remain to be answered when building QoS enabled assistive environments. Typical QoS strategies that affect user-perceived quality or fidelity of an application generally do not apply to context-aware pervasive applications and for this reason, it is necessary to devise other mechanisms and metrics to define, establish, and guarantee QoS. In the context of an assistive environment, we observe QoS from two different aspects, including the QoS that the system can provide to the end users and the QoS that the various system components can provide to each other. Both aspects are equally important for the smooth and unobtrusive operation of an assistive environment.

Many QoS attributes need to be defined in a way that is specific to the features they are applied to. The following are two major QoS attributes that are applied to the system level:

- *Consistency*: Consistency can be defined as the ability of the system to maintain a standard behavior in the type and the end-to-end delay of the output given to the user as well as the ability of the different system components to obey certain constraints that are imposed to achieve a certain level of operation quality, regardless of the changes that might occur in the environment. The consistency of a system can be measured as the number of times that the constraints were not obeyed and by what degree.
- *Adaptability*: Adaptability refers to the ability of the system to adapt to varying workloads regarding number of users to serve, number of simultaneous events, the varying resource demands of an application, and the increasing amounts of data to be stored and processed while preserving any previously established agreements on service performance. The adaptability can be measured, for example, as the percentage increase in the amount of workload that the system can tolerate without losing its stability and consistency compared to the initially estimated workload. The percentage increase can be defined as:

$$\frac{newWorkload - initialWorkload}{initialWorkload} \times 100$$

5.2.7 Cost

The cost of an assistive environment must be controlled so that it is affordable to its user base. An assistive environment typically consists of many software and hardware components. The way in which those components are integrated can significantly affect the overall performance, and thus must be managed carefully. In par-

ticular, the most expensive components put together may not always deliver the best performance system-wide. Cost can also be controlled by seeking a balance between optimal performance and affordability. Note that the cost of an assistive environment does not only include the purchase price, but also the cost of maintenance.

The proposed framework identifies the following major attributes to be used for cost evaluation:

- *Installation Cost*: This is the cost that has to be paid to set up an assistive environment. It includes both the cost of purchasing the necessary hardware and software components, and the cost of putting them together and installing them in the physical space.
- *Maintenance Cost*: Maintenance activities are often necessary to keep an assistive environment up and running. Examples of such activities include regular replacement of sensor batteries, system reset after a breakdown, hardware and software components upgrade, and such.

Note that both installation and maintenance costs contribute to the overall cost of an assistive environment. In addition, there is often an interplay between the two types of cost. For example, some sensors cost more but are more robust and have a longer lifespan, which reduces the cost of maintenance in the long run. Therefore, the two types of cost should be considered in an integrative manner.

5.3 Discussion

Assistive environments are unique in that they target a special group of users who are typically not familiar with technologies. Thus, user adoption is the key to the success of those environments. In this work, we have identified a set of attributes that are considered critical to user adoption. Table 5.1 summarizes those attributes. The framework also suggests sample metrics, as well as possible approaches to mea-

Table 5.1. Summary of attributes to evaluate a Human-Centered Computational Framework for Assistive Living.

General Attributes	**Specific Attributes**
Functionality	- Correctness - Robustness - Reliability
Usability	- Ease of Use - Accessibility - Non-obtrusiveness
Security and Privacy	- Violation reports - Configurable privacy/access control - Encryption strength
Architecture	- Modularity - Interoperability
Intelligence	- Accuracy - Precision - Recall or Sensitivity - Specificity
Quality of Service (QoS)	- Consistency - Adaptability
Cost	- Installation Cost - Maintenance Cost

suring them, to quantify those attributes. In case the proposed system does not meet requirements, the designer has to go back and improve things to meet the minimum defined standards. This work is part of a larger effort in building an infrastructure for evaluating assistive environments. The infrastructure will consist of hardware components that are commonly needed to deploy an assistive environment, and a collection of software tools that help to automate the evaluation process. We plan to build a database of operation and user profiles that are representative of real life scenarios that may occur in assistive environments. Those profiles will provide us with a more realistic assessment of those environments.

CHAPTER 6

CONCLUSION

In this dissertation, I have presented my work in creating a Computational Framework for Human-Centered Multimodal Data Analysis. The proposed framework examines different views of an assistive environment to support the human well being by providing services to improve health condition and quality of life. We have examined what it takes to set up such an environment, how to collect the necessary data to understand the human condition and behavior and how to efficiently analyze the collected data in order to obtain accurate results regarding the underlying problem.

Our findings show that the same basic computational methods can be used to analyze different aspects of the human presence inside an assistive environment, such as behavioral patterns and health condition. Efficient feature selection methods allow us to reduce the size of the problem and successfully extract meaningful information out of data collected from various sources. We have suggested methods to tackle specific problems, such as cancer and sleep problems detection, which can generalize to other similar human-centered activities or conditions. Furthermore, we have shown that the fusion of heterogeneous data about the same subject coming from different sources can improve the accuracy of the obtained results and we have suggested methods to efficiently do so.

Finally, we have examined the various aspects that would make a human-centered computational framework built for assistive environments successful in real life and we have suggested metrics to quantitatively measure and and evaluate those

aspects. Table 6.1 summarizes the problems framework that our proposed tried to solve and the methods to solve them.

Table 6.1. Summary of problems solved by our Human-Centered Computational Framework and methods we proposed to solve them.

Problem	**Suggested Methods**
Human-Centered Data Collection	- *Sleep Patterns*: Pressure mat, Kinect - *Medication*: RFID reader - *Sensor placement/coordination*: Factor Graphs - *Longitudinal events*: Detection of Episodes - *QoS*: Ontology centered middleware
Feature Selection and Analysis of Human-Centered Data	*Various Cancer Types - aCGH data* - Hybrid Sparsity Regularization for feature selection - Accurate Cancer classification - Identification of Biomarkers (disease-related genes)
Human-Centered Data fusion	*Brain Tumors* - Fusion of Gene Expression & MRS data - Accurate Brain Tumor Typing - Identification of Biomarkers (disease-related genes) *Sleep patterns* - Fusion of pressure and depth data - Sleep Pattern Recognition
System Evaluation	*Evaluation of Computational Framework* - Identification of Important Attributes to Evaluate - Proposition of Quantitative Evaluation Metrics

REFERENCES

[1] A. Jaimes, D. Gatica-Perez, N. Sebe, and T. S. Huang, "Guest editors' introduction: Human-Centered Computing–Toward a human revolution," *Computer*, vol. 40, no. 5, pp. 30–34, 2007.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.

[3] T. S. Rappaport and S. B. O. (Firme), *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[4] S. Abiteboul, R. Hull, and V. Vianu, "Foundations of databases," 1995.

[5] C. M. Bishop and S. S. e. ligne), *Pattern recognition and machine learning*. springer New York, 2006, vol. 4.

[6] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[7] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[8] K. S. Fu, R. C. Gonzalez, and C. S. Lee, *Robotics: control, sensing, vision, and intelligence*. McGraw-Hill, Inc., 1987.

[9] "Kinect-xbox.com," http://www.xbox.com/en-US/kinect.

[10] E. Becker, V. Metsis, R. Arora, J. Vinjumur, Y. Xu, and F. Makedon, "SmartDrawer: RFID-based smart medicine drawer for assistive environments." ACM, 2009, p. 49.

[11] A. Papangelis, V. Metsis, J. Shawe-Taylor, and F. Makedon, "Sensor placement and coordination via distributed multi-agent cooperative control," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2010, pp. 1–8.

[12] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings, "Decentralised coordination of continuously valued control parameters using the max-sum algorithm," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 601–608.

[13] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings, "Decentralised coordination of low-power embedded devices using the max-sum algorithm," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 639–646.

[14] K. Park, Y. Lin, V. Metsis, Z. Le, and F. Makedon, "Abnormal human behavioral pattern detection in assisted living environments," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2010, pp. 1–8.

[15] R. Arora, V. Metsis, R. Zhang, and F. Makedon, "Providing QoS in ontology centered context aware pervasive systems." ACM, 2009, p. 8.

[16] A. J. Vander, J. H. Sherman, D. S. Luciano, M. J. Kluger, and L. G. d'Alecy, *Human physiology: the mechanisms of body function*. McGraw-Hill New York etc., 1990.

[17] L. G. Astrakas, D. Zurakowski, and A. A. Tzika, "Noninvasive magnetic resonance spectroscopic imaging biomarkers to predict the clinical grade of pediatric brain tumors," *Clin Cancer Res*, vol. 10, pp. 8220–8228, 2004.

[18] L. L. Cheng, D. C. Anthony, A. R. Comite, P. M. Black, A. A. Tzika, and R. G. Gonzalez, "Quantification of microheterogeneity in glioblastoma multiforme with ex vivo high-resolution magic-angle spinning (HRMAS) proton magnetic resonance spectroscopy," *Neuro-Oncology*, vol. 2, no. 2, p. 8795, 2000.

[19] D. Morvan, A. Demidem, J. Papon, M. D. Latour, and J. C. Madelmont, *Melanoma Tumors Acquire a New Phospholipid Metabolism Phenotype under Cystemustine As Revealed by High-Resolution Magic Angle Spinning Proton Nuclear Magnetic Resonance Spectroscopy of Intact Tumor Samples 1.* AACR, 2002, vol. 62, no. 6.

[20] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, *et al.*, *Gene Expression-based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification 1.* AACR, 2003, vol. 63, no. 7.

[21] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed, "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, p. 249, 2003.

[22] K. Y. Kim, J. Kim, H. J. Kim, W. Nam, and I. H. Cha, "A method for detecting significant genomic regions associated with oral squamous cell carcinoma using aCGH," *Medical and Biological Engineering and Computing*, vol. 48, no. 5, pp. 459–468, 2010.

[23] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. L. Kuo, C. Chen, and Y. Zhai, "High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays," *Nature genetics*, vol. 20, pp. 207–211, 1998.

[24] C. F. Aliferis, D. Hardin, and P. P. Massion, "Machine learning models for lung cancer classification using array comparative genomic hybridization." American Medical Informatics Association, 2002, p. 7.

[25] K. Chin, S. DeVries, J. Fridlyand, P. T. Spellman, R. Roydasgupta, W. L. Kuo, A. Lapuk, R. M. Neve, Z. Qian, and T. Ryder, "Genomic and transcriptional aberrations linked to breast cancer pathophysiologies," *Cancer cell*, vol. 10, no. 6, pp. 529–541, 2006.

[26] A. Daemen, O. Gevaert, K. Leunen, E. Legius, I. Vergote, and B. D. Moor, "Supervised classification of array cgh data with hmm-based feature selection." World Scientific Pub Co Inc, 2009, p. 468.

[27] T. Gambin and K. Walczak, "A new classification method using array comparative genome hybridization data, based on the concept of limited jumping emerging patterns," *BMC bioinformatics*, vol. 10, no. Suppl 1, p. S64, 2009.

[28] J. Huang, A. Salim, K. Lei, K. O'Sullivan, and Y. Pawitan, "Classification of array CGH data using smoothed logistic regression model," *Statistics in medicine*, vol. 28, no. 30, pp. 3798–3810, 2009.

[29] C. Lengauer, K. W. Kinzler, and B. Vogelstein, "Genetic instabilities in human cancers," *Nature*, vol. 396, no. 6712, pp. 643–643, 1998.

[30] J. Liu, S. Ranka, and T. Kahveci, "Classification and feature selection algorithms for multi-class CGH data," *Bioinformatics*, vol. 24, no. 13, p. i86, 2008.

[31] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J. J. Daudin, "A statistical approach for array cgh data analysis," *BMC bioinformatics*, vol. 6, no. 1, p. 27, 2005.

[32] D. Pinkel and D. G. Albertson, "Array comparative genomic hybridization and its applications in cancer," *Nature Genetics*, vol. 37, pp. S11–S17, 2005.

[33] S. Riccadonna, G. Jurman, S. Merler, S. Paoli, A. Quattrone, and C. Furlanello, "Supervised classification of combined copy number and gene expression data," *Journal of Integrative Bioinformatics*, vol. 4, no. 3, p. 74, 2007.

[34] H. Willenbrock and J. Fridlyand, "A comparison study: applying segmentation to array cgh data for downstream analyses," *Bioinformatics*, vol. 21, no. 22, p. 4084, 2005.

[35] E. S. Venkatraman and A. B. Olshen, "A faster circular binary segmentation algorithm for the analysis of array CGH data," *Bioinformatics*, vol. 23, no. 6, p. 657, 2007.

[36] R. Tibshirani and P. Wang, "Spatial smoothing and hot spot detection for CGH data using the fused lasso," *Biostatistics*, vol. 9, no. 1, p. 18, 2008.

[37] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[38] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," *Advances in neural information processing systems*, vol. 19, p. 41, 2007.

[39] G. Obozinski, B. Taskar, and M. Jordan, "Multi-task feature selection." Citeseer, 2006.

[40] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Machine Learning Proceedings of the Fifteenth International Conference (ICML'98)*. Citeseer, 1998, pp. 82–90.

[41] A. Y. Ng, "Feature selection, l_1 vs. l_2 regularization, and rotational invariance." ACM, 2004, p. 78.

[42] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," *NIPS 2010*, 2010.

[43] N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ Pr, 2000.

[44] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*. Wiley-Interscience, 2000, vol. 354.

[45] S. L. Cessie and J. C. V. Houwelingen, "Ridge estimators in logistic regression," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 41, no. 1, pp. 191–201, 1992.

[46] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[47] D. Luo, C. Ding, and H. Huang, "Towards structural sparsity: an explicit ℓ_2/ℓ_0 approach," *ICDM 2010*, 2010.

[48] Y. Wang and F. Makedon, "Application of Relief-F feature filtering algorithm to selecting informative genes for cancer classification using microarray data," p. 498, 2004.

[49] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization." Morgan Kaufmann Publishers, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.32.9956

[50] 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*, vol. 11, no. 1, pp. 10–18, 2009.

[51] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[52] C. Yau, V. Fedele, R. Roydasgupta, J. Fridlyand, A. Hubbard, J. W. Gray, K. Chew, S. H. Dairkee, D. H. Moore, and F. Schittulli, "Aging impacts transcriptomes but not genomes of hormone-dependent breast cancers," *Breast Cancer Res*, vol. 9, no. 5, p. R59, 2007.

[53] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," 2001.

[54] V. Metsis, O. C. Andronesi, H. Huang, M. N. Mindrinos, L. G. Rahme, F. Makedon, and A. A. Tzika, "Combination of sparse and wrapper feature selection from multi-source data for accurate brain tumor typing," *ISMRM 2011*, 2011.

[55] V. Metsis, H. Huang, F. Makedon, and A. Tzika, "Heterogeneous data fusion to type brain tumor biopsies." Springer, 2009, p. 233.

[56] V. Metsis, D. Mintzopoulos, H. Huang, M. N. Mindrinos, P. M. Black, F. Makedon, and A. A. Tzika, "Multi-Source feature selection to improve multi-class brain tumor typing," *ISMRM 2009*, 2009.

[57] J. M. Legler, L. A. Ries, M. A. Smith, J. L. Warren, E. F. Heineman, R. S. Kaplan, and M. S. Linet, "Brain and other central nervous system cancers: recent trends in incidence and mortality," *Journal of the National Cancer Institute*, vol. 91, no. 16, p. 1382, 1999.

[58] A. A. Tzika, L. Astrakas, H. Cao, D. Mintzopoulos, O. C. Andronesi, M. Mindrinos, J. Zhang, L. G. Rahme, K. D. Blekas, A. C. Likas, *et al.*, "Combination of high-resolution magic angle spinning proton magnetic resonance spectroscopy and microscale genomics to type brain tumor biopsies," *INTERNATIONAL JOURNAL OF MOLECULAR MEDICINE*, vol. 20, no. 2, p. 199, 2007.

[59] O. C. Andronesi, K. D. Blekas, D. Mintzopoulos, L. Astrakas, P. M. Black, and A. A. Tzika, "Molecular classification of brain tumor biopsies using solid-state magic angle spinning proton magnetic resonance spectroscopy and robust classifiers," *International journal of oncology*, vol. 33, no. 5, p. 1017, 2008.

[60] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," *LECTURE NOTES IN COMPUTER SCIENCE*, p. 171171, 1994.

[61] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, 1995, p. 88.

[62] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes," in *Third Conference on Email and Anti-Spam (CEAS)*. Citeseer, 2006, pp. 125–134.

[63] V. N. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 988–999, 1999.

[64] A. A. of Sleep Medicine, *International classification of sleep disorders, revised: Diagnostic and coding manual*, 2001.

[65] D. Foley, S. Ancoli-Israel, P. Britz, and J. Walsh, "Sleep disturbances and chronic disease in older adults: Results of the 2003 national sleep foundation sleep in america survey," *Journal of Psychosomatic Research*, vol. 56, no. 5, pp. 497 – 502, 2004.

[66] M. Vandeputte and A. de Weerd, "Sleep disorders and depressive feelings: a global survey with the beck depression scale," *Sleep Medicine*, vol. 4, no. 4, pp. 343 – 345, 2003.

[67] A. E. Flores, J. E. Flores, H. Deshpande, J. A. Picazo, X. Xie, R. Franken, H. C. Heller, D. A. Grahn, and B. F. O'Hara, "Pattern recognition of sleep in rodents using piezoelectric signals generated by gross body movements," *IEEE transactions on bio-medical engineering*, vol. 54, no. 2, p. 225, 2007.

[68] B. H. Jansen and W.-K. Cheng, "Classification of sleep patterns by means of markov modeling and correspondence analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 5, pp. 707 –710, 1987.

[69] J. V. Marcos, R. Hornero, D. Alvarez, F. del Campo, and C. Zamarron, "Assessment of four statistical pattern recognition techniques to assist in obstructive

sleep apnoea diagnosis from nocturnal oximetry," *Medical engineering & physics.*, vol. 31, no. 8, p. 971, 2009.

[70] W. Liao and C. Yang, "Video-based activity and movement pattern analysis in overnight sleep studies," *Proceedings*, vol. 3, no. Conf 19, pp. 1774–1777, 2008.

[71] H. Ni, B. Abdulrazak, D. Zhang, S. Wu, Z. Yu, X. Zhou, S. Wang, and I. conference; 7th, "Towards non-intrusive sleep pattern recognition in elder assistive environment," 2010.

[72] Sabri Boughorbel, Fons Bruekers, and Jeroen Breebaart, "Baby-Posture Classification from Pressure-Sensor Data," *ICPR*, pp. 556–559, 2010.

[73] Z. Jin, Z. Lou, J. Yang, and Q. Sun, "Face detection using template matching and skin-color information," *Neurocomputing*, vol. 70, no. 4-6, pp. 794–800, 2007.

[74] G. H. Dunteman, *Principal components analysis*. Sage Publications, Inc, 1989.

[75] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[76] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, Feb. 1989.

[77] B. W. Boehm, J. R. Brown, and M. Lipow, "Quantitative evaluation of software quality," in *Proceedings of the 2nd international conference on Software engineering*. IEEE Computer Society Press, 1976, pp. 592–605.

[78] A. F. Newell and P. Gregor, "Design for older and disabled peoplewhere do we go from here?" *Universal Access in the Information Society*, vol. 2, no. 1, pp. 3–7, 2002.

[79] M. Y. Ivory and M. A. Hearst, "The state of the art in automating usability evaluation of user interfaces," *ACM Computing Surveys (CSUR)*, vol. 33, no. 4, pp. 470–516, 2001.

[80] I. Korhonen and J. E. Bardram, "Guest editorial introduction to the special section on pervasive healthcare," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 3, pp. 229–234, 2004.

[81] M. Shaw and D. Garlan, "Software architecture: perspectives on an emerging discipline," 1996.

[82] C. Y. Baldwin and K. B. Clark, *Design rules: The power of modularity*. The MIT Press, 2000, vol. 1.

[83] N. Lavrac, P. Flach, and B. Zupan, "Rule evaluation measures: A unifying view," *Inductive Logic Programming*, pp. 174–185, 1999.

[84] S. Shenker, "Specification of guaranteed quality of service," 1997.

[85] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 7, pp. 1228–1234, 1996.

[86] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li, "QoS-aware middleware for ubiquitous and heterogeneous environments," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 140–148, 2001.

[87] J. Jin and K. Nahrstedt, "QoS specification languages for distributed multimedia applications: A survey and taxonomy," *Multimedia, IEEE*, vol. 11, no. 3, pp. 74–87, 2004.

BIOGRAPHICAL STATEMENT

Vangelis Metsis received his Computer Science B.Sc. degree from the Department of Informatics of Athens University of Economics and Business in 2005. During 2006-2007 he worked as a research associate at the National Center for Scientific Research "Demokritos" in Athens, Greece before joining the Heracleia Human-Centered Computing Laboratory at UTA to work as a research assistant towards his Ph.D. degree. His research interests include Machine Learning, Data Mining, Bioinformatics and Pervasive Computing. Mr. Metsis has co-authored several peer reviewed papers published in technical conferences and journals and has served as a committee member and reviewer in many others. Mr. Metsis is currently a member of Upsilon Pi Epsilon Texas Gamma Chapter, and Golden Key honor societies.