Physical Activity Recognition in Daily Life using a Triaxial Accelerometer

ILLAPHA CUBA GYLLENSTEN

Master of Science Thesis
Stockholm, Sweden 2010
Physical Activity Recognition in Daily Life using a Triaxial Accelerometer

ILLAPHA CUBA GYLLENSTEN

Master’s Thesis in Biomedical Engineering (30 ECTS credits) at the School of Computer Science and Engineering Royal Institute of Technology year 2010
Supervisor at CSC was Erik Fransén
Examiner was Anders Lansner

TRITA-CSC-E 2010:061
ISRN-KTH/CSC/E--10/061--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc
Abstract

Activity recognition with a tri-axial accelerometer worn at the waist could provide an objective measure of physical activity. Current methodology usually trains and evaluates the activity recognition algorithms on laboratory collected data with the goal ultimately being to infer these activities under daily life conditions with free-living subjects.

This thesis examines the effects of this by evaluating three popular machine learning algorithms - Artificial neural networks, Decision trees and Support vector machines - trained on a large set of laboratory data in both a laboratory trial and in a daily-life trial. The golden standard used to measure the activities is the multi-sensor activity monitor IDEEA. It is found that accuracies of all algorithms employed (best SVM: 81.9%, best ANN: 81.5 %, best decision tree: 82.6%) drop significantly during the daily-life trials (best SVM: $-13.4 \pm 14.4\%$, p<0.05 best ANN: $-15.3 \pm 15.9\%$ p<0.05, best Decision tree: $-20.3 \pm 12.8\%$ p<0.05), as compared to a laboratory trial weighted to represent the distribution of daily life activities. However, by applying a meta-classifier adapting the outputs to the activity patterns found in daily-life a substantial improvement in classification accuracy is achieved ($+9.1 \pm 7.4\%$ compared to SVM).

Referat

Klassificering av fysiska aktiviteter i dagliga livet med hjälp av en triaxial accelerometer

Accelerometerdata från en midjeburen triaxial accelerometer skulle kunna användas för att mäta fysisk aktivitet objektivt. Nuvarande metodologi använder data insamlade under laboratorielika förhållanden för att träna maskininlärnings algoritmer att känna igen dessa aktiviteter. Förhoppningen är att sedan kunna använda dessa algoritmer för att bestämma fysisk aktivitet på personer under fria former i dagliga livet.

Detta examensarbete undersöker detta tillvägagångssätt genom att utveckla algoritmer från tre populära maskininlärningsparadigmer – Artificial neuronnät, Stödvektormaskiner och beslutsträd – vilka tränas med en stor mängd data inhämtat i laboratorium för att sedan evalueras med data från laboratoriet och från dagliga livet. Den gyllene standarden som används i dagliga livet är IDEEA, en kommersiell apparat för aktivitetsigenkännning användandes flera sensorer.

Testen visar att säkerheten i klassificeringen (bästa SVM: 81.9%, bästa ANN: 81.5 %, bästa besluts trädet: 82.6%) sjönk påtagligt i dagliga livet (bästa SVM: $-13.4 \pm 14.4\%$, p<0.05 bästa ANN: $-15.3 \pm 15.9\%$ p<0.05, bästa besluts trädet: $-20.3 \pm 12.8\%$ p<0.05). Genom att använda en meta-klassificerare som tränades för att känna igen aktivitetsmönster så kunde prestandan ökas ($+9.1 \pm 7.4\%$ jämfört med SVM).
Acknowledgements

The seven months of research and work leading to the finalization of this master thesis was conducted at the Philips Research Laboratories in Eindhoven, Holland. The author wishes to thank all the support received, without which this thesis would never have been.

First and foremost, I would like to thank my supervisor at Philips, Alberto Bonomi, who helped me at virtually every step of the thesis, from the experimental design to proof reading the thesis. His insight and experience in the field helped greatly in my research.

I’m also grateful for the help I have received from Erik Fransén, my academic supervisor, who helped me with structuring the thesis and directing me to the academic research that has been conducted in the field.

I would also like to extend my thanks to the many interns and coworkers at Philips Research, which provided assistance in my research and, more importantly, for showing me such a great time in Holland.

Finally, I would like to thank Philips Research itself for providing the fundings for this project and for its commitment to primary and fundamental research.
Contents

1 Introduction ............................... 1
  1.1 Scope and Limitations .......................... 2
  1.2 Aim .................................. 3
  1.3 Method ................................ 4

2 Background ................................. 5
  2.1 Accelerometry ............................ 5
    2.1.1 Accelerometer data from physical activity .................................. 6
    2.1.2 Early research in activity recognition ........................................ 7
  2.2 Machine learning theory ................... 7
    2.2.1 Artificial neural networks .................. 7
    2.2.2 Decision trees .......................... 10
    2.2.3 Support vector machines .................. 12

3 Construction of the classifiers .......... 14
  3.1 Classifying time-series data .............. 14
    3.1.1 Choosing an appropriate window-length .................. 15
    3.1.2 Digital signal processing .................. 16
    3.1.3 Commonly considered features .................. 17
    3.1.4 Features extracted from the windowed data .................. 18
    3.1.5 Feature normalization .................. 21
    3.1.6 Feature selection .................. 21
  3.2 Measuring classifier performance .......... 23
    3.2.1 Cross Validation .................. 23
    3.2.2 Accuracy, F-score and the confusion matrix .................. 23
  3.3 Support vector machine ................ 24
    3.3.1 Multi-label SVM’s .................. 25
  3.4 Artificial neural networks ................ 25
  3.5 C4.5 algorithm .......................... 25
  3.6 Meta-algorithms ......................... 26
    3.6.1 Plurality voting .................. 26
    3.6.2 Bootstrap aggregating .................. 26
    3.6.3 Fusing information from several windows .................. 26
## CONTENTS

4 Experimental results 28

4.1 Experimental design 28
4.2 Training data 29
  4.2.1 Lab data set: 1 29
  4.2.2 Lab data set: 2 30
  4.2.3 Lab data set: 3 31
  4.2.4 Training and optimization on the three data sets 32
4.3 Testing data 34
  4.3.1 Study population data 35
  4.3.2 Comparing the outputs of the models in daily life with the golden standard. 36
  4.3.3 Performance of laboratory trained on the laboratory collected data 36
  4.3.4 Performance of laboratory trained models on daily-life data 37
4.4 Analysis 38
  4.4.1 Weighing performance by its distribution in daily-life. 38
  4.4.2 Training on daily-life data 40

5 Discussion 41
  5.1 Results from the laboratory data 41
  5.2 Results from the daily life data 42
  5.3 Summary and Conclusions 43
  5.4 Limitations 43
  5.5 Outlook and future work 44

Bibliography 45

Appendices 48

A Features 49

B Figures 53

C Selected code 58
Chapter 1

Introduction

The field of activity recognition is concerned with automatically identifying types of physical activity from a set of measurements. These measurements could contain data from devices such as video cameras, heart rate monitors, pedometers, goniometers and accelerometers. The activity types are often so easily classified by humans visually that little or no confusion arises from verbs such as: running, jumping, walking, falling, sleeping etc. While easily classified by humans, automatic detection has yet to achieve a comparable amount of classes and accuracy. The advantage of having such a recognition system is obvious; direct observation of visual data is expensive and very time consuming and recording subjects for long periods of time requires a controlled environment.

Alternatively, subjects could be told to keep a diary of all the activities they undertake, noting the start and stop time of each activity. Several standardized diaries exists, for example see Wolf et al.[1], but the procedure is prone to recall bias and subjectiveness, it also inconvenient for the subjects[2].

The need for objective physical activity data has arisen from a number of fields, especially in health related research. With the shift in lifestyle occurring in the modern world, where work and leisure tend to be less physically demanding, research on the health effects of low physical activity is necessary. Several reports have already found links between common diseases and physical inactivity, for example: cardiovascular diseases, obesity, diabetes mellitus, hypertension, cancer and depression[2]. The importance of physical activity has led it to be considered a leading health indicator by “Healthy People 2010”, a U.S. governmental organization promoting public health and providing a framework for groups working with preventive medicine[3]. Programs designed to increase physical activity could make major contributions to public health and it is therefore important to properly evaluate such initiatives with an objective and reliable system.

Another health related topic is assessing functional ability, which has a large impact on quality of life and is a determining factor for independent living. Currently it is assessed with a variety of tools such as: photogrammetry, kinematic and kinetic analyses, video recording, electromyography, force plate analysis, simple-timed
measures, questionnaire tools, validated functional tests and observation. These methods are often time consuming and require a controlled environment[4]. Activity recognition could be used to record data over activities and transition times e.g. time spent transitioning between sitting and standing. The data could be collected unobtrusively, automatically and without needing a controlled environment therefore it could provide doctors with daily-life data of parameters that indicate the functional ability of a patient. This is valuable when assessing the need of placing a patient under supervised care or to evaluate to what degree a medication increases a patient’s functional ability.

With an aging population in the industrialized world the care taking of the elderly is becoming ever more important. Falling is one of the major hazards for the elderly and the length of time spent unaided strongly affects the rate of morbidity[5]. Activity recognition could be used to automatically detect a fall and alarm emergency workers. This could help to reduce the time spent unaided, possibly lowering the amount of fall-related deaths in the elderly.

Managing metabolic diseases could be a complicated endeavor and a valuable measure helping patients with this is the total energy expenditure[6]. After the resting energy expenditure, energy expenditure due to physical activity is the largest part of the total energy expenditure. Previous research has shown that there exists a strong correlation between this energy expenditure and rectified signal intensity (RSI\(^1\)) during a specific activity type[2]. If the specific activity type could be determined from the data then this information could be used when estimating the energy expenditure. This might improve the accuracy of energy expenditure estimation from wearable accelerometers and thereby making it easier to manage metabolic diseases.

Not all topics using activity recognition are health-related. For example, ubiquitous computing is a model for human-computer interaction where information from various everyday objects are integrated and processed. In this model users engage several computing devices simultaneously, possibly without being aware of doing so. An example of a pervasive computing system would be a user engaging an activity recognition device that recognizes that the user is sleeping and therefore automatically turns off the users reading lamp. Examples of ubiquitous computing systems using activity recognition can be found in Randell[7] and Weiser[8].

These examples highlight the reasons why physical activity classification has become an active interdisciplinary research topic, with contributions from fields such as medicine, computer science and signal processing.

### 1.1 Scope and Limitations

In this thesis accelerometer data is used to recognize physical activity types. This approach has several advantages. First and foremost it is a proven method, reports dating back to the eighties have used accelerometers for activity recognition[9]. Sec-

\(^{1}\)RSI is a simple measure of the variability of an accelerometer signal.
1.2. AIM

ondly, advancements in MEMS\textsuperscript{2} technology has made these accelerometer sensors available at low cost in a miniaturized and low-power format while still achieving a reliable output\cite{4}.

When constructing the activity recognition system no consideration has been taken in respect to the computational complexity of the algorithms employed. Though this is a very important aspect for on-board applications where the activity recognition is done in real-time, it is less so in devices which uploads its data to be analyzed.

A review article on activity recognition by Preece et al.\cite{2} concluded that in most studies comparing classifier performance both Decision trees and Artificial neural networks have been shown to outperform other algorithms. However, SVM’s have usually not been considered in such comparisons and have shown promise in small scale studies\cite{10}\cite{2}.

To restrict the study to a few machine learning algorithms it was decided that these popular and previously successful algorithms would be studied, namely: Artificial Neural Networks, Support Vector Machines and Decision trees.

1.2 Aim

Unobtrusiveness is an important characteristic in applications were activity recognition is integrated into the daily life of the intended users. If the users are supposed to wear a monitor, which is the case for accelerometers, it would be preferable if it is small and light. Unfortunately, making devices small and light constrains the maximum battery capacity, leading to shorter times between recharging, which is an important aspect for applications with prolonged observation.

Using only a single location with a device carrying a single sensor has advantages for unobtrusive designs since it consumes less energy and does not need communication between sensors located at different body parts. This communication would either need: wires, which could be perceived as obtrusive; or a wireless transmitter and receiver, leading to higher power consumption.

However, using a single site considerably reduces the informational content which implies that recognition performance will be reduced, but to what extent this happens is not clear.

This begs the question: To what extent does a single-site accelerometer compare to a state-of-the-art multi-sensor?

While no study has previously addressed this particular question studies have been conducted using several sensors, comparing the accuracies for different sensor combinations. An example of such a study is the Bao-Intille 2004 study\cite{11}. In their study they used 5 accelerometers placed on the right hip, left thigh, right ankle, left arm and the right wrist, they found that using a single accelerometer produced losses in accuracy of 30% compared to using all accelerometers. However their study was done in a semi-naturalistic environment and considered a large set of activities (20).

\textsuperscript{2}Microelectromechanical systems also known as micro systems technology or MST.
CHAPTER 1. INTRODUCTION

The importance of using data from daily-life settings is recognized by several researchers\cite{2}\cite{12} and while efforts could be made to try and mimic the conditions under which daily activities takes place, subjects would still be constrained to specific tasks.

In light of this the following research questions were formulated:

- How accurate is a waist mounted accelerometer when compared to a multi-sensor device in daily life?
- Which machine learning algorithm is best suited for detection of some common activities in daily life using a waist mounted accelerometer?

1.3 Method

The algorithms were developed for the Tracmor\cite{Philips Research, Noord-Brabant, Netherlands} a single-site tri-axial accelerometer sampling at 20 Hz mounted around the waist. Training was done on a large set of data collected in a laboratory environment.

As previously stated the width of classifications currently possible in activity recognition does not compare to human visual analysis. However, for a small subset of activities or groups of broad activities, activity recognition using accelerometers have shown considerable promise\cite{13}\cite{11}. Therefore the algorithms were trained to distinguish between 6 broad activity classes thought to be common in daily-life:

- Lying down: resting or sleeping on a bed or sofa.
- Sitting: working in front of computer, sitting at a table or in a car.
- Standing: standing still or standing doing light household chores such as cleaning or doing the dishes.
- Walking: including walking upstairs and downstairs.
- Cycling.
- Running.

In daily-life the classifications were compared to those made by IDEEA\textsuperscript{\textregistered}\cite{Mini Sun, California, United States}, a commercially available activity recognition device using sensors placed under the sole of the feet, the thighs and the chest. The IDEEA has been validated to very high accuracies\cite{14}\cite{15} but since there are some activities which it does not recognize an activity diary was also provided to the subjects.

The IDEEA and the activity diary form the golden standard in daily-life. The reasoning behind choosing such an unusual golden standard lies in the impracticality of manually observing free-living subjects. Recording subjects in daily-life is also riddled with privacy issues and it is not clear how such a protocol could be constructed.
Chapter 2

Background

This chapter provides a background describing the instrumentation used to record the physical activity and briefly the theory behind the machine learning algorithms subsequently used to categorize it.

2.1 Accelerometry

The activity recognition device used in this study, the Tracmor (see figure B.5), houses three micro-machined accelerometers pointing in orthogonal directions i.e. a tri-axial accelerometer.

The working principle behind one of these accelerometers is that of a damped mass-spring system, where the output corresponds to the distance between the suspended mass and a reference point. The reference point is usually chosen so that 0 corresponds to a free-fall condition and that the maximum output number corresponds to the maximum amount of $g$ that the device can register. Figure 2.1 shows such a conceptualized accelerometer.

![Figure 2.1. An accelerometer resting on a surface subject to the reaction force of gravity. The working principle is illustrated by a suspended mass and a readout scale, reading: -1 g.](image)
The device shown in the figure has an output of $-1 \text{g}$ resting in its depicted orientation. If it were turned on its side, placing the spring perpendicular to the gravity vector, the output would read $0 \text{g}$. The accelerometer would also output $0 \text{g}$ if it were thrown up in the air, since by construction a free-fall condition corresponds to an output of $0 \text{g}$.

These thought experiments serve to show that the output of an accelerometer can contain information about the orientation of the device in respect to the gravitational reference frame, and that it can contain information about the dynamic movements of the device. However, in general the contributions from these two sources can not be distinguished from one another.

### 2.1.1 Accelerometer data from physical activity

For classification purposes the muddling of orientation and movement might not be too problematic as human motion often produce quite characteristic signals, see figure 2.2.

![Figure 2.2. A typical recording from the accelerometer showing 8 minutes of recorded data during a supervised test where the subject is asked to perform specific tasks. See figure 4.1 for the definitions of the positive directions of the accelerometer](image)

Indeed, simple thresholding of measures such as variance and mean have been employed successfully for simple classifications in a laboratory environment.

For more complex classifications these simple measures might not suffice. To ensure that useful information is captured when extracting more intricate measures, specific knowledge of the characteristics of acceleration signals generated by human movement is necessary.
2.2. MACHINE LEARNING THEORY

A study by Cappozo[16] on acceleration data arising from physical activity found that when walking at natural velocity the bulk of the acceleration power spectrum lies between 0.9-5 Hz in the upper body. Bhattacharya et al.[17] found that during jumping, frequencies measured at the back were contained between 0.78 - 4 Hz depending on jump height. Using force plate analysis Sun and Hill[18] found that the major energy band during daily activities was between 0.3-3.5 Hz. Though frequencies of up to 60 Hz have been recorded at the ankle these findings seem to suggest that, at the waist, frequencies below 6 Hz will include most of the information expected to be found in daily life signals. Digital signal processing schemes could then be employed to remove content not generated by physical activity, most likely resulting from noise.

2.1.2 Early research in activity recognition

Early research using accelerometer data to classify human motion used simple accelerometers having piezo-electric elements which where only able to measure changes in the proper acceleration. Referring back to the conceptualized accelerometer, figure 2.1, the output would be the difference in the mass position with respect to time. From this data a simple measure called counts was extracted, this is calculated by taking the absolute value of the output and integrating it over a specific time-period:

\[ \text{count} = \int_{t_0}^{t_0+\Delta} |a(t)| dt. \]

For example, a study by Bouten et al.[19] manually adjusted thresholds of the measured counts to divide the activities into sedentary, moderate and vigorous. Though this was somewhat successful in estimating energy expenditure the approach lacked the sophistication to be applied to more complex classifications such as that between cycling and walking. In recent years a more systematic approach to the classification of physical activity has taken form, borrowing techniques from the vast field of machine learning and statistical modeling.

2.2 Machine learning theory

The following sections show how some machine learning algorithms infer rules for classification from a set of given examples. For a more thorough treatment the reader is referred to: “Fundamentals of Neural Networks” [20], “An Introduction to Support Vector Machines and other kernel-based learning methods” [21] and “Programs for Machine Learning” [22].

2.2.1 Artificial neural networks

Artificial neural networks (ANN) is a computational model that emulates, to a certain extent, the learning systems found in nature. Composed by many small computational units, all sharing the same fundamental design, they mimic the structure
of a nervous system. The computational units further emulate nature by resembling biological neurons receiving input from their input nodes, not unlike dendrites, and transmitting one value to other computational units, not unlike the axon of a neuron.

The approach is very flexible and a wealth of different structures has been formed by connecting and organizing these units together. This flexibility has enabled this model to be applied to conceptually very different types of problems e.g. clustering data, compressing data and as a tool to understand the human brain [20].

Activity recognition is often posed as a supervised learning problem were patterns or inputs are provided with associated activity types or labels. The goal of the network is to associate the labels with the patterns. A reasonable network structure for this type of problem is the multi-layered feed-forward neural network, see figure 2.3. Previous research has successfully applied this design in activity recognition [13][2]. From the figure one can distinguish three distinct layers: the input layer which forwards the data to be categorized, the hidden layer which is solely a computational layer and the output layer, which perform the final classifications. Although this figure shows just a single hidden layer, in general a feed-forward network could contain any number of hidden layers.

![Figure 2.3. A feed-forward network with one hidden layer. The arrows indicate from where the data comes from (a computational unit, or from input data) and to where it goes (as input to a computational unit or as output of the whole network structure). The computational units are represented by a circle.](image)

The output from one single computational unit is calculated by passing the weighted sum of all inputs plus a bias term to an activation function:

$$\text{output} = f(w_0 + \sum_i x_i \cdot w_i)$$
2.2. MACHINE LEARNING THEORY

where \( x_i \) is the value of an input, \( w_i \) is the corresponding weight for that input and \( w_0 \) is the bias term. The function \( f \) is usually a monotonic smooth function chosen so that the range of the output varies between two finite values e.g. -1 and 1. Examples include: algebraic functions, the arctangent, the error function and the hyperbolic tangent, see figure 2.4.

![Activation functions](image.png)

**Figure 2.4.** Activation functions \( f \) for different sigmoid shaped functions normalized to have a derivative of 1 at 0 and a range between -1 and 1. Legend: Black, error function; Red, the hyperbolic tangent; Blue, an algebraic function, \( x/(\sqrt{x^2 + 1}) \); Green, the arctangent function.

The relation between output and input for a single unit can be changed by adapting the weights and the bias term. It can be shown that by changing the weights for a whole network structure of units any input to output relation can be modeled\[23\]. This highlights the versatility of neural networks. However, the interest in these models has been sparked by their ability to learn from given examples.

In order to learn from the examples the network needs a measure for how well it is performing i.e a cost function for the errors. A reasonable choice of cost function for pattern recognition problems is the mean squared error, defined as:

\[
\text{error} = \frac{1}{2} \sum (y_i - c_i)^2
\]

where \( y_i \) is the output of unit \( i \) in the output layer and \( c_i \) is the correct output for this example, in classification problems usually 1 or 0. This error can be backpropagated through the network and using optimization techniques such as conjugate gradient descent or the Levenberg-Marquardt algorithm\[24\][25], this error can be minimized.

An important realization is that all optimization techniques incrementally adapt the weights after evaluating the solution in a neighborhood around the current weights. Therefore the network could converge to different solutions for the same set of examples depending on how the weights are initialized\[20\]. This property makes ANNs an *unstable* learning algorithm.
2.2.2 Decision trees

A decision tree classifier uses a tree-like graph to separate the input data into distinct classes, see figure 2.5. The classifier uses simple rules at each step to separate the data into smaller and smaller sets e.g. dividing the data into those examples having a green color and those that do not. Eventually the data is split into many groups containing suggested classifications.

Examples of algorithms that construct these type of classifiers are: CART, C4.5 and C5. They differ between how they choose the rules on which to split the data and how they decide when to stop splitting to avoid overly complex structures.

One of the more popular methods of generating decision trees is the C4.5 algorithm introduced by Quinlan in 1993, it is chosen in this study because of its previous successes in activity recognition[11] and its non proprietary nature\(^1\).

The algorithm works by splitting the data using the most informative attribute rule not yet considered in the path from the root. To measure informativeness it borrows the concept of information entropy from information theory. The information entropy measures the amount of information carried by a random discrete variable and is defined as:

$$\text{information entropy of } x = - \sum_i P(x = x_i) \log_b(P(x = x_i)),$$

where \(P(x = x_i)\) denotes the probability of the random discrete variable \(x\) assuming the value \(x_i\) and \(b\) is a constant typically chosen to normalize the maximum

\(^1\)the updated and more modern version C5 is a licensed software
2.2. MACHINE LEARNING THEORY

entropy to 1. Consequently, a high information entropy corresponds to a uniform probability distribution of \( x \) and a low information entropy corresponds to a very peaked probability distribution.

Treating the class label as a random discrete variable, so that the probability \( P(x = x_i) \) is estimated as the ratio of class \( i \) in the training set, it is possible to calculate the information entropy of the training examples. Comparing this entropy to the weighted average of the entropy of the sets generated by a split, a value of the “informativeness” can be calculated. The figure 2.6 shows a computation of this value for the attribute rule “is the color green?”. This calculation is quick and inexpensive thus it can easily be applied to all the possible splits, the split carrying the highest information is therefore quickly identified as the most informative rule. This method can then be recursively applied at each node in the tree, expanding it until no reduction in entropy is possible.

![Figure 2.6](image)

Weighted average of the entropy after the split = 0.73.

**Figure 2.6.** The figure shows the distribution of the fruits before and after the split with displayed entropy. The average is weighted by the number of examples (fruits) the set contains e.g. for the green fruits this is 13/34 and for the non-green fruits it is 21/34. Using this attribute rule i.e. is the color green? a reduction of \( 0.95 - 0.73 = 0.22 \) in entropy is achieved, comparing this for all attribute rules a decision can be made on which one is the most informative.

This is the general idea behind the C4.5 algorithm, however the informational gain is modified since otherwise attributes which have many different entries tend to be favored. Furthermore, since the algorithm only handles discrete values other rules are invoked to discretize continuous values. Heuristics for when to stop splitting the nodes are also included in the complete algorithm as the tree produced by splitting until no more splitting can be done is generally a very complex structure.
2.2.3 Support vector machines

First developed in the late seventies\cite{26} support vector machines, or SVM, have received a great deal of attention from the machine learning community. It has by now developed a strong mathematical foundation and rigorous statistical analysis, which could be contrasted by the previous methods which rely on heuristics or analogies of human learning.

The idea behind SVMs is to find the plane which maximizes the margin between the input data of two classes, see figure 2.7. To find this plane it is first recognized that any plane in any dimension can be expressed with the following formula:

\[ \vec{w} \cdot \vec{x} - b = 0 \]

where \( \vec{w} \) is the normal of the plane formed by all points \( \vec{x} \) satisfying the equation. Consequently, any two non-identical parallel planes can be written as:

\[ \vec{w} \cdot \vec{x} - b = 1 \]
\[ \vec{w} \cdot \vec{x} - b = -1 \]

as seen in the figure 2.7. Restricting the input data of the two classes so that the classes are separated by the two planes can be expressed with the following constraints:

\[ \vec{w} \cdot \vec{x}_i^+ - b \geq 1 \quad \forall \ i \]
\[ \vec{w} \cdot \vec{x}_j^- - b \leq -1 \quad \forall \ j \]
2.2. MACHINE LEARNING THEORY

where $\vec{x}_i^+$ is the input vector of a positive example $i$ and $\vec{x}_j^-$ is the inputs of a negative example $j$.

Linear algebra reveals that the distance between the two planes is inversely proportional to the norm of the vector $\vec{w}$, or $||\vec{w}||$. Therefore, finding the maximum margin hyperplane can be formulated as an optimization problem i.e. minimize $||\vec{w}||$ subject to the aforementioned constraints. Interestingly this problem is strictly convex i.e. it has only one global minimum, making the classifier a stable machine learning algorithm.

Evidently, this classifier can only be produced if there exists a plane that perfectly splits the data. Absent of such a plane, the algorithm will not work. However, by relaxing the constraints one could allow for misclassified examples. The formulation below presents the relaxed constraints of the optimization problem, with one introduced slack variable for each example:

\[
\vec{w} \cdot \vec{x}_i^+ - b \geq 1 - \xi_i^+
\]
\[
\vec{w} \cdot \vec{x}_j^- - b \leq -1 + \xi_j^-.
\]

Associated with the slack variables is a cost for introducing them, since having fewer misclassified examples is better. The combined cost of misclassification and that of having a large margin can be combined while conserving the convex property of the algorithm[27]. This soft-margin approach introduces a tunable variable $C$ which determines how much misclassified examples are penalized in the trade-off between small slack variable values and a large margin.

The classifier is still a linear classifier and as such it might not be able to model more complex feature dependencies. However, it is possible to represent some non-linear relations linearly in a high-dimensional space. This allows the former theory for linear classifications to be applied while producing non-linear boundaries in the input data space. The mapping of the vector product in this high dimensional space can easily be computed by using a kernel function. An example of a kernel function is the radial-basis function:

\[
\text{RBF}(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2).
\]

Using this kernel introduces a second tunable parameter $\gamma$. By tuning these parameters it is possible to represent different types of input-output relations while still having the statistical stability of the SVM approach.
Chapter 3

Construction of the classifiers

This chapter explains the details behind the construction and optimization of the classifiers.

3.1 Classifying time-series data

Classifying accelerometer signals into physical activities falls under the general problem of discrete time-series classification. In general, associating time-varying data with classes presents problems for machine learning algorithms inferring rules from a set of examples, where each example is considered separately as an input vector with an output vector.

This assumption, that separating the examples produces no informational loss, requires careful construction of the input vector. Importantly, the relevant temporal information included in the time-series has to be accounted for.

The most common way to account for this temporal information in activity recognition[2] is to use a “sliding-windows” approach. This approach can take many forms but can be split into two steps:

- First the time-series data is split into a number of “windows” all containing a small part of the data:

  \[
  \text{window}_i = [\text{data}(t_i), \text{data}(t_i + \delta t), ..., \text{data}(t_i + l \cdot \delta t)]
  \]

  where \( \delta t \) is the time between two sampled frames and \( l \) is an integer indicating the number of frames in the window. This window is then “slided” an amount \( \Delta t \) to form the next window, see figure 3.1:

  \[
  t_{i+1} = t_i + \Delta t
  \]

  This amount of sliding also determines the resolution of the classification, which will all be of width \( \Delta t \).

- The second step in the approach is to calculate a set of features from each window. This feature set is then used as the input vector for the example
3.1. CLASSIFYING TIME-SERIES DATA

window. The features could be the Fourier components of the window, the mean value of the windowed data or any other measure derived from the discrete data in the window.

3.1.1 Choosing an appropriate window-length

For activity recognition where accelerometer data from physical activity is windowed, the choice of the number of frames, \( l \), is guided by a trade-off between two aspects, information and resolution.

If the subject is performing a single activity, then a long window will include more information about that activity. The increased amount of information could make activity separation easier, thus making the classifier stronger. This effect favors long windows and improvements of activity recognition performance has been observed with windows spanning 45.4 seconds (2048 frames) for some particular activities\[28\]. It should be noted that this is not always the effect of increasing the window size, indeed for some activities a reduction in recognition accuracy with longer windows can be observed.

The preceding discussion, favoring long windows, is based on that the windowed data only contains a single activity, which is how laboratory data usually is collected (long bouts of a single activity). However, the assumption that the window, no matter how long, will only contain a single activity is likely to be violated in daily-life where activity changes are uncontrolled. If the window contains more than one activity, classification is ambiguous at best, and more likely impossible. Logically a longer window will have a higher probability of containing several activities than a shorter one, thus this effect favors short windows.

Unfortunately, to optimize the trade-off between the first and the second effect, data over activity durations in daily life has to be collected. For a general population data over time spent in various activities is unavailable, presumably because of the cost of direct observation, thus only a general guideline was adopted; if performance is not significantly increased with a longer window, choose the shorter one.

The first effect i.e. the improvement of activity recognition as a function of the window length, has been studied previously\[28\][29]. According to Tapia\[28\], who used the C4.5 algorithm to distinguish between a large set of activities (52) with 5 sensors, the accuracy obtained using FFT components leveled off at a window size of 5.6 seconds (256 frames). Using another feature, the accumulated absolute sum, the performance leveled off at a window length of 22.7 seconds (1024 frames). Ultimately, Tabia decided to use a 5.6 second window. Another study conducted by Bonomi\[29\], also using the C4.5 algorithm with only a single sensor and less activities, found that performance did not improve with windows larger than 6.4 seconds (128 frames).

These two previous studies suggest that little can be gained with sizes exceeding 6.4 seconds. Taking the previous guideline into account it was decided to use windows of only 6.4 seconds, this having the extra benefit of containing 128 time frames which, having a radix of 2, makes for simple frequency calculations.
CHAPTER 3. CONSTRUCTION OF THE CLASSIFIERS

Another consideration is the amount of jump between consecutive windows. Choosing $\Delta t$ to one frame would generate the most amount of training examples. However, features extracted between two consecutive window will not contain much new information and will increase the developing time of the algorithms. Previous research in activity recognition has successfully employed 50% overlap between adjacent windows\cite{11}\cite{30} which has also been chosen for this study thus $\Delta T$, is chosen to be $l/2$, figure 3.1 shows such a segmentation.

![Figure 3.1. Windowing of accelerometer data into sections with 50% overlap. The resolution of classification is therefore 50% of the window size. The convention chosen in this study is to center the classifications around the windows.](image)

3.1.2 Digital signal processing

Before splitting the data into windows and calculating features the signal could be prepared to only contain physically relevant information. As explained in section 2.1.1 physical activity usually generates frequency components between 0.3 and 6 Hz. Information in the signal outside of this band can be removed by applying bandpass and lowpass filters, in effect attenuating noise and unwanted information. A bandpass and a lowpass filter was developed with MATLAB and used when processing of the data.

**Windowing filters**

When calculating the Fourier transform of a short window the components with wavelengths that do not divide the window size in integers will cause artifacts. This
3.1. CLASSIFYING TIME-SERIES DATA

effect is known as spectral leakage and to diminish this effect one usually employs a windowing function. The trade-off of doing this is that the frequency-resolution diminishes but lower amplitude signals will be discernible, see B.1 for an example. Since the features calculated bin together components in groups of 1.25 Hz the frequency-resolution is not a major issue. A comprehensive treatment of optimal window function is somewhat out of the scope of this thesis, however a reasonable choice of a windowing function is the Hamming window. This window has a side-lobe width of approximately 0.2 Hz which is more than adequate considering the bin sizes (the side-lobe width determines the frequency-resolution). The figure 3.2 illustrates the superiority of the Hamming window in attenuating frequencies further away than 0.2 Hz from the main lobe.

![Figure 3.2. Graph showing the Fourier transform for the hamming and the rectangular window. The side-lobe width is greater for the hamming window though overall attenuation is better for the hamming window.](image)

3.1.3 Commonly considered features

After filtering and windowing the signal features were extracted from the signal. In a review article by Preece et al.\cite{2} covering different strategies of activity recognition four groups of features that previously have been employed by researchers were identified:

Heuristic features

This is comprised of features derived from an intuitive or fundamental understanding of how a movement or a posture will produce a characteristic signal. An
example of this is the angular orientation of the device, which can be derived if the subject is still.

**Time-domain features**

These are features derived directly from the window, and usually of a statistical nature. This category includes features such as mean, variance, skewness, kurtosis, cross-correlation etc.

**Frequency-domain features**

These are features derived from the Fourier transform of the windowed data. These features include the specific frequency components, the spectral energy, spectral entropy and cross-spectral density.

**Time-Frequency domain features**

These are features derived using transforms which include both spatial and temporal information. The most common and famous one is the wavelet transform.

### 3.1.4 Features extracted from the windowed data

A large feature set (113 features) was constructed using heuristic, time-domain and frequency features. The features did not include wavelets, although they have been employed successfully in activity recognition[31][32], a recent comparison study[33] suggest that: for short time-windows and healthy young subjects wavelets might not perform any better or even worse than Fourier components combined with simple time measures. A suggested reason for this is the high periodicity of accelerometer data in short time windows.

The bulk of the considered features have been used in other studies[29][2][28] therefore only a few features will be given a more thorough description, for the complete list of features see appendix A.

**Angle between the gravitational vector and the transverse plane**

This feature represents the angle formed between the gravitational vector and the plane formed by the the dorsoventral axis and mediolateral axis, see figure 3.3 and has previously been used by[34]. Intuitively this would be 0 if the subject is lying down, and around 90 degrees if the person is standing, or sitting. Of course, this feature only represents angle when the subject is reasonably still. While mean values in the x-axis could also differentiate between standing and lying down, the angle feature would consistently show 0 for subjects regardless of how they are lying down, stomach, side or back. A histogram of the discriminative ability of this feature extracted from the training data is shown in figure 3.4. The calculation of the angle is done under the assumption that the accelerometers do not have offsets.
3.1. CLASSIFYING TIME-SERIES DATA

i.e. that the magnitude of the output for a motionless accelerometer does not depend on its orientation. This assumption is generally not true for physical accelerometers which due to technical limitations will suffer from offsets.

![Diagram showing the angle between the gravity vector and the plane formed by the dorsoventral axis and the mediolateral axis for two postures (sitting and lying down).](image)

**Figure 3.3.** Diagram showing the angle between the gravity vector and the plane formed by the dorsoventral axis and the mediolateral axis for two postures (sitting and lying down).

![Histogram plot of angle for different activities.](image)

**Figure 3.4.** Three histograms showing the mean angle value for the windows of three sedentary postures (standing, sitting and lying down) extracted from the training data. The Y-axis indicate the normalized frequency of occurrence i.e. for each activity the bars will sum to one. It can be seen that, for the training data, this feature easily separates lying from the other activities.
CHAPTER 3. CONSTRUCTION OF THE CLASSIFIERS

Motionlessness

With a single waist mounted accelerometer it might be difficult to distinguish between standing still and sitting down. However, intuitively it could be expected that when people are standing they shift their weight or sway to a greater extent than when they are sitting down. The standard deviation of acceleration data can show this, however if the person is quickly shifting their position while sitting the window might be misclassified. A measure that might be more robust to this type of error is the longest period of no acceleration change, or motionlessness. To account for low amplitude noise one can employ a tolerance gap, within which the acceleration data can vary while still being considered “motionless”. For a simple MATLAB script calculating this see appendix C. Unlike the previous feature this feature does not produce a histogram in which classes are easily separated, instead it provides some information on which class might be more likely for different values, see figure 3.6.

![Figure 3.5. Two histograms showing the longest sequence of frames without motion in the vertical direction for the windows of two sedentary postures (standing and sitting) extracted from training data. The Y-axis indicate the normalized frequency of occurrence i.e. for each activity the bars will sum to one. It can be seen that this feature carries some information of whether the person is sitting or standing.](image)

Peak frequencies

Some activities produce quite clear peaks in the Fourier-spectra and extracting features at these points lead to valuable features, see 3.6. To extract features at these points a clear definition of what constitutes a peak needs to be formulated. In this thesis a simple peak measure is used: a positive peak of prominence $k$ is defined as a datapoint, $d_i$, for which the $k$ adjacent values i.e. $[d_{i-k}, ..., d_{i-1}, d_{i+1}, ..., d_{i+k}]$ are smaller.
3.1. CLASSIFYING TIME-SERIES DATA

Figure 3.6. Three histograms showing the amplitude of the peak frequency in the vertical direction for the windows of three dynamic activities (cycling, walking and running) extracted from the training data. The Y-axis indicate the normalized frequency of occurrence i.e. for each activity the bars will sum to one. This feature readily separates running from the two other activities, and to a great extent cycling from walking.

3.1.5 Feature normalization

The features constructed all have different magnitudes, some larger and some smaller than the others. This can cause problems for some of the machine learning algorithms, where features having a higher magnitude will, a priori, be given a higher emphasis. Applying a normalizing step before training can counter-act this unwanted effect.

The normalization procedure chosen rescales all features so that they have a zero mean and unit variance, for an example \( \hat{x}_j \) and the feature \( i \) the rescaled example \( x_{j,i} \) would be:

\[
x_{j,i} = \frac{\hat{x}_{j,i} - \text{mean}_j(\hat{x}_{j,i})}{\text{std}_j(\hat{x}_{j,i})}.
\]

The function \( \text{mean}_j() \) and \( \text{std}_j() \) are short-hand notations for the mean and standard deviation over all the values index \( j \) can assume, in this case it is the value of the feature \( i \) averaged over all examples \( j = 1, 2, ..., n \) where \( n \) is the number of existing examples.

3.1.6 Feature selection

The complete feature set included 113 different characteristics. Having such a large set of features may cause redundancy which can result in an unnecessary increase in computational complexity and may be detrimental for some learning algorithms[35]. A feature selection procedure applied before training could reduce redundancy and
complexity and might make better classifiers. Therefore, two extra feature sets were constructed with two simple feature selection procedures.

**Manually selected features**

This feature set was constructed by manually inspecting features in the training data with histograms and selected on an intuitive understanding of how they might help in this particular problem. The selection procedure was more intuitive than thorough owing to the sheer size of features available. The selected features are marked with a star “∗” in the appended feature list, see appendix A.

**Automatic selection**

Following an idea presented by Chen and Lin[36], features were selected on the basis of how well they could separate the data linearly. To illustrate the working principle consider a binary problem with two classes (positive (+) and negative (-)), the score of each feature is then defined as:

\[
\text{score of feature } i = \frac{(\text{mean}_j(x_{j,i}^+) - \text{mean}_j(x_{j,i}))^2 + (\text{mean}_j(x_{j,i}^-) - \text{mean}_j(x_{j,i}))^2}{\text{std}_j(x_{j,i}^+)^2 + \text{std}_j(x_{j,i}^-)^2}.
\]

where \(x_{j,i}^+\) is value of feature \(i\) of an example \(j\) having a positive class label, +, and \(\text{std}_j()\), \(\text{mean}_j()\) are shorthand notations for the mean and standard deviation calculated over the index \(j\), i.e. over the examples.

Features having a large distance between the mean of the two classes in comparison with the standard deviation of each class, will get a high score and are likely to have a high discriminative ability. Therefore, a ranking of the features could easily be obtained with the score.

It should be noted that some features that may, in conjunction with other features, be very discriminative could score low since no feature dependencies are calculated. This neglect of feature dependency can also result in the top ranked features being very similar, causing redundancy. Nevertheless, the method is simple, easy to implement, fast and has been employed successfully[37][36].

The paper that introduce this selection method had SVMs in mind for the implementation. SVMs are binary classifiers and therefore the method is easily adapted to them. When developing the multi-label SVMs in this thesis each binary SVM used the top 10 ranked features. For other machine learning algorithms that are not binary the selection method can be adapted using the same techniques that are used to make multi-label SVMs. In this study a one-against-all voting technique is used, it can be described by the following steps:

1. Let the first class be regarded as the positive class and all other classes as negative. Calculate the top 10 scoring features.
2. Repeat the first step, replacing the positive class with the second class, then the third class and so on until all classes have been covered.

3. collect the top scoring features for every run of step one and remove any duplicate features, the resulting feature list is used for classification.

### 3.2 Measuring classifier performance

To evaluate the performance of the different classification algorithms researchers usually calculate the accuracy of the algorithm i.e. the percentage of correctly classified examples. Care has to be taken when calculating this and any other performance measure on training data so that over fitted models score low.

Overfitting occurs when the learning algorithms creates a mapping that is highly optimized for the training data which, being overly tuned to the idiosyncrasies of the training data, gives poor performance when applied to new and unseen data. A common technique employed to guard against over fitted models scoring high is cross-validation.

#### 3.2.1 Cross Validation

The idea behind most validation techniques is to train the algorithm on a subset of the data and then evaluate it on the part of the data not seen during training. This is how a cross-validation procedure starts, but when the training and evaluation is done, it switches a chunk of the training data with the evaluation data and re-trains the algorithm. This procedure is then iterated until all the data has been trained and evaluated on. The average of the performance measure for all iterations is the cross-validated estimate of the algorithm.

Arguably the most common way of splitting the data is \( n \)-fold cross-validation.

In \( n \)-fold cross-validation data is split randomly into \( n \) sets of approximately equal size. In each iteration the algorithm is trained on \( n - 1 \) sets and evaluated on the left out set. Common values for \( n \) is: all samples, also called leave-one-out, 10 and 5.

In activity recognition the algorithms are going to be applied to unknown subjects for which no training has been conducted therefore it could be worthwhile to try and get rid of subject specific biases. In Leave-one-subject-out cross validation or LOSO, the left out chunks are the entire collection of samples from a single subject. Since the training data consists of 52 subjects this procedure is computationally more complex than a 5 or 10 fold cross-validation.

#### 3.2.2 Accuracy, F-score and the confusion matrix

To measure the performance of the algorithms this thesis uses four different metrics: The confusion matrix, the accuracy and the F-score. The most informative metric of these three is the confusion matrix, consisting of a table comparing all classifica-
tions to the true class, for an example see B.1. Indeed, the two other measures can be computed from this table. However, the amount of information included in the table makes the measure difficult to comprehend, and cluttered, thus it is used only sparingly when other measures, are not sufficient. A simpler measure is accuracy, it is defined as all correctly identified examples divided by the total number of examples. This measure is easy to understand and ubiquitous in activity recognition, all referenced articles about activity recognition employs this measure.

For multi-class problems a more in depth analysis taking into account class specific performances may be appropriate. Consider a multi-class problem with errors defined with respect to the class $A$, represented by the following table:

<table>
<thead>
<tr>
<th></th>
<th>classifier outputs $A$</th>
<th>classifier outputs any other class</th>
</tr>
</thead>
<tbody>
<tr>
<td>true class is $A$</td>
<td>true positive, $tp$</td>
<td>false negative, $fn$</td>
</tr>
<tr>
<td>true class is any other class</td>
<td>false positive, $fp$</td>
<td>true negative, $tn$</td>
</tr>
</tbody>
</table>

Defining accuracy for a single class as the true positives divided by the sum of true positives and false negatives, neglects the false positives of the outcome. This measure can therefore not differentiate between a classifier being overly sensitive to a class and one having high discriminative ability for that class.

The F-score is a performance measure that takes this effect into account by putting equal emphasis on finding the true classes i.e. single class accuracy, called recall in information retrieval, and being discriminative called precision i.e. not returning many false negatives.

\[
\text{precision} = \frac{tp}{tp + fp} \quad \text{recall} = \frac{tp}{tp + fn}
\]

The harmonic mean of these two measures form the F score, which as accuracy varies between 0 and 1:

\[
\text{F-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

### 3.3 Support vector machine

To develop the support vector machines the package LibSVM\[38\] was added to the software suite MATLAB. The SVMs used the radial basis kernel and the soft margin penalty for misclassifications, as explained in section 2.2.3. To optimize the two tunable variables, one for the radial basis radius ($\gamma$) and one for the penalty term ($C$), a grid-search was performed on different combinations. The grid search evaluated 70 combinations, where $C$ ranged between $[2^{-1}, ..., 2^8]$ and $\gamma$ ranged between $[2^{-5}, ..., 2^1]$. The strength of each combination was estimated by performing a 5-folded cross-validation.
3.4. ARTIFICIAL NEURAL NETWORKS

3.3.1 Multi-label SVM’s

As the standard formulation of a SVM is a binary classifier some technique has to be applied to achieve multi-label classification. A simple approach is to train several binary SVM’s and combine their outputs into a single classification. Two common methods are:

- one-against-all This approach trains $n$ classifiers, one for each label. Each single SVM is trained with one class marked as positive and all other classes treated as negative. The class of the SVM with the highest decision value is chosen as the output of the complete classifier.

- one-against-one This approaches trains $\frac{n(n-1)}{2}$ classifiers, where each classifier is trained in distinguishing between two classes. The class that has been predicted by the majority of the classifiers is considered to be the output of the complete classifier. To resolve ties a the class with the lowest index gets priority.

3.4 Artificial neural networks

The Neural networks were developed with MATLAB, using the Levenberg-Marquardt algorithm to minimize the weights and early stopping to avoid over fitting. The implementation of early stopping used 75 % of the data for minimizing the weights and 25 % to validate the net, the training continued as long as the mean-squared-error did not worsen 6 consecutive times in a row. When the training was stopped, the weights and biases having the lowest validation error was used.

To optimize the performance of the neural network a search was conducted to find the best number of hidden neurons in the hidden layer. The 5-folded cross-validated performance was evaluated for each configuration 10 times since the selection of folds and the starting configuration of the weights is random. If a neural net with fewer neurons was able to achieve comparable performance to a more complicated model, the smaller one was chosen. This was estimated by visually inspecting the F-score for the classifiers.

3.5 C4.5 algorithm

In section 2.2.2 it was mentioned that decision trees use heuristics to decide when to stop splitting the nodes further. These simple heuristic techniques may result in overly complex and over fitted trees. A method to reduce this complexity is pruning i.e fusing nodes into leaves to create a smaller tree. This will remove long chains of rules that might not carry much informative content into a single classification, scoring lower on performance but the tree might be more general.

By plotting the cross-validated performance of each level of pruning, the simplest tree delivering good performance could be found. This was also estimated by visually inspecting the performance plots.
CHAPTER 3. CONSTRUCTION OF THE CLASSIFIERS

3.6 Meta-algorithms

In a recent study comparing classifier performance, meta-level schemes that combined different techniques achieved the best performance [30]. This was also mentioned in Preece’s review article [2]. Here a few common methods for combining classifier performance are described: bootstrap aggregating, plurality voting and lastly a method to weigh temporal dynamics on daily-life data is introduced.

3.6.1 Plurality voting

Plurality voting aims at combining the output of several classifiers to reduce errors that are committed by the single classifiers. This implementation in this thesis gives each classifier equal weight in voting for the class, the class with most votes becomes the output of the meta-classifier. In the event of a tie the classification is awarded to the class with the lowest index.

3.6.2 Bootstrap aggregating

One of the considered classifiers, ANN, is unstable in the sense that, given a training data set it will not always converge to the same values for weights and biases. For example, [20] describes the vastly different behavior of two neural nets trained on the same data.

Stabilizing the output might be achieved by training several ANN’s and aggregating their output. Varying what data is used as training data during classifier generation might further improve generalization.

A technique to achieve this introduced by Breiman [39] is bagging, or bootstrap aggregating. This technique generates m number of classifiers, each one trained on a slightly different data set \( D_i \). The data set is randomly drawn, with replacement, from the pool of all available training data \( D \). If the sizes of \( D_i \) and \( D \) is the same, each classifier would on average be trained on 63.2% of the unique data present in \( D \). Breiman [39] suggest that aggregating more than 25 classifiers might not improve accuracy for classifiers with few classes, and that improvement started to level off at 10 classifiers, which is the number chosen for this study.

3.6.3 Fusing information from several windows

Since activities in consecutive timeframes are thought to be correlated e.g. a person cycling is probably not going to be lying down in the next time frame. It could be worthwhile to try and include past and future classifications when classifying the current time-frame.

A simple meta-classifier approach is to employ an ANN that weighs classification data from several timeframes to determine the most probable output of the timeframe to be classified. For this purpose the following network structure is proposed:

This structure is chosen so that the weights can be initialize to make the network work as a plurality voter; only considering classifications from the current timeframe.
3.6. META-ALGORITHMS

Figure 3.7. An ANN constructed to weigh the classifications of both several classifiers (plurality voting) and in time (temporal fusion). This example shows the weighing of six classifiers from 5 time-frames and with six classes. The hidden neurons each specialize in detecting one activity for one time-frame, they receive as input the a binary value indicating which classifiers detected the activity for which the neuron is specialized. The output layer weighs the contributions from all the hidden neurons to produce a classification for t=0.

This ensures that the training of the net can start with a reasonable solution to the problem. The structure also ensures that net will use reasonable and fairly simple rules since the hidden neurons only work as plurality voters. Training will then adapt the weights of the net so that more weight is given to classifiers that are better at recognizing certain activities, and that the information of temporal dynamics between the classes is included from previous and future time-frames.
Chapter 4

Experimental results

4.1 Experimental design

To record physical activity the device containing the tri-axial accelerometer (the Tracmor) was mounted on the lower back of each subject, see figure 4.1.

Figure 4.1. Schematic drawing of the tri-axial accelerometer showing placement on the lower back and the three directions in which accelerometer data is recorded. The arrows point in the positive direction of each axis, resulting in a left-handed orientation.

Models were developed using the techniques described in chapter 3 with data from three previously conducted experimental trials using this mounting location.
and device. The section 4.2 describes the performance and details of that data in greater detail. When the construction of the models was done they were evaluated on a new set of laboratory collected data and set of data collected from free-living subjects in daily life, see section 4.3. Finally, a meta-classifier using outputs of each model was trained with the daily-life data this is described in subsection 4.3.3.

4.2 Training data

The Tracmor has been used at Philips previously for research, both in activity recognition studies and studies in energy expenditure estimation. Data for activity recognition has been collected by Alberto Bonomi and a previous master student, Chen Xin. What follows is the activities found in these training sets, some data of the study population and the method used to label it.

4.2.1 Lab data set: 1

This dataset contained recordings from 22 Lean subjects recruited by Alberto Bonomi but because of sensor malfunction or misalignment 2 had to be removed and the remaining 20 were used for the training. Biometric data for the remaining subjects show that the group has a moderate BMI, length and age distribution.

<table>
<thead>
<tr>
<th></th>
<th>men</th>
<th>women</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>13</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>age [years]</td>
<td>23 − 43 (29.2 ± 7.4)</td>
<td>23 − 35 (27.9 ± 5.2)</td>
<td>23 − 43 (29.2 ± 6.7)</td>
</tr>
<tr>
<td>weight [kg]</td>
<td>60.6 − 92.4 (74.4 ± 8.0)</td>
<td>53 − 79 (66.1 ± 9.9)</td>
<td>53 − 92.4 (71.5 ± 9.4)</td>
</tr>
<tr>
<td>height [m]</td>
<td>1.68 − 1.97 (1.79 ± 0.08)</td>
<td>1.56 − 1.74 (1.66 ± 0.06)</td>
<td>1.62 − 1.97 (1.74 ± 0.1)</td>
</tr>
<tr>
<td>BMI [kg/m²]</td>
<td>20.2 − 28.1 (23.3 ± 2)</td>
<td>19.2 − 32.6 (24.1 ± 5)</td>
<td>19.2 − 32.6 (23.6 ± 3.2)</td>
</tr>
</tbody>
</table>

During the experiment each subject was asked to perform a specific activity under the supervision of the experimenter. For each activity that was performed the experimenter would note the start and stop time thus creating a timestamp that later was used to isolate segments containing a single activity. The following list show the specific activities performed, the label used for training the classifiers and the average amount of time that the activity was recorded (averaged over all subjects).

- lying down on a bed, labeled as: lying down, average segment size: 283.5 [s].
- sitting still in a chair, labeled as: sitting, average segment size: 144 [s].
- sitting in a chair while working on a computer, labeled as: sitting, average segment size: 199.9 [s].
• standing still, labeled as: standing, average segment size: 145.65 [s].

• standing over a sink doing dishes, labeled as: dynamic standing, average segment size: 146.6 [s].

• walking, labeled as: walking, average segment size: 47.55 [s].

• walking up a staircase, labeled as: walking, average segment size: 129.3 [s].

• walking down a staircase, labeled as: walking, average segment size: 140.8 [s].

• walking at 4 different paces, all labeled as: walking, average segment size, respectively: 188.4, 182.6, 136.8, 133.2 [s].

• running at 4 different paces, all labeled as: running, average segment size, respectively: 104, 102.5, 73.8, 63.4 [s].

• cycling at 4 different paces, all labeled as: cycling, average segment size, respectively: 119.6, 115.2, 84.9, 81.2 [s].

The total time of all segmentations amount to 14 hours and 34 minutes, using the windowing method described in 3.1.1 this equals 15840 examples.

4.2.2 Lab data set: 2

This dataset, also collected by Alberto Bonomi, contained accelerometer data from 21 obese subjects of which 19 had correctly mounted sensors, these subjects were subsequently used for training. The biometric data shows that these subjects had very high BMI and weight though other characteristics were normal.

<table>
<thead>
<tr>
<th></th>
<th>men</th>
<th>women</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>6</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>age [years]</td>
<td>44 − 65 (56.9 ± 8.4)</td>
<td>42 − 70 (55.8 ± 9.2)</td>
<td>42 − 70 (56.2 ± 8.7)</td>
</tr>
<tr>
<td>weight [kg]</td>
<td>83.4 − 182.6 (113 ± 33.0)</td>
<td>69.4 − 129.1 (84.3 ± 16.3)</td>
<td>69.4 − 182.6 (94.9 ± 27)</td>
</tr>
<tr>
<td>height [m]</td>
<td>1.69 − 1.84 (1.76 ± 0.06)</td>
<td>1.49 − 1.76 (1.65 ± 0.08)</td>
<td>1.49 − 1.84 (1.69 ± 0.09)</td>
</tr>
<tr>
<td>BMI [kg/m²]</td>
<td>28.5 − 53.9 (36.1 ± 8.5)</td>
<td>24.9 − 41.7 (30.9 ± 4.5)</td>
<td>24.9 − 53.9 (32.8 ± 6.6)</td>
</tr>
</tbody>
</table>

The activities performed were recorded using the same procedure as in data set 1, albeit with a different list of activities.

• standing still performed 5 times between other activities, labeled as: standing, average segment size, respectively: 53.6, 48.1, 47.8, 69.6, 47.5 [s].

• sitting still in a chair, labeled as: sitting, average segment size: 69.2 [s].
4.2. TRAINING DATA

- sitting in a chair working on a computer, labeled as: sitting, average segment size: 102.9 [s].

- doing the dishes then sweeping the floor, labeled as: dynamic standing, average segment size: 113.4 [s].

- walking at 2 different paces, labeled as: walking, average segment size, respectively: 113.4, 113 [s].

- cycling (not performed by all subjects), labeled as: cycling, average segment size: 117.3 [s].

The segments of this set amount to 4 hours and 18 minutes, or approximately 5540 training examples.

4.2.3 Lab data set: 3

This dataset contained recordings of 14 subjects recruited by Chen Xin, a previous master student, who used this to detect sport activities in a gymnasium. Out of these 14 subjects one subject had to be removed due to sensor misalignment. The remaining 13 were used for training.

<table>
<thead>
<tr>
<th></th>
<th>men</th>
<th>women</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>10</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>age [years]</td>
<td>26 - 49</td>
<td>25 - 30</td>
<td>25 - 49</td>
</tr>
<tr>
<td></td>
<td>(35 ± 8.9)</td>
<td>(27.3 ± 2.5)</td>
<td>(32.9 ± 8.2)</td>
</tr>
<tr>
<td>weight [kg]</td>
<td>65 - 99</td>
<td>53 - 63.5</td>
<td>53 - 99</td>
</tr>
<tr>
<td></td>
<td>(77.5 ± 10.8)</td>
<td>(57.5 ± 5.4)</td>
<td>(74.2 ± 13.4)</td>
</tr>
<tr>
<td>height [m]</td>
<td>1.68 - 1.89</td>
<td>1.64 - 1.78</td>
<td>1.64 - 1.89</td>
</tr>
<tr>
<td></td>
<td>(1.81 ± 0.07)</td>
<td>(1.69 ± 0.08)</td>
<td>(1.78 ± 0.08)</td>
</tr>
<tr>
<td>BMI [kg/m^2]</td>
<td>20.7 - 30.6</td>
<td>19.7 - 20.3</td>
<td>19.7 - 30.6</td>
</tr>
<tr>
<td></td>
<td>(23.8 ± 3.3)</td>
<td>(20.0 ± 0.3)</td>
<td>(23.2 ± 3.4)</td>
</tr>
</tbody>
</table>

The activities were performed under supervision of the experimenter and included the following activities, all isolated segments contained 60 [s] of data:

- sitting still in a chair, labeled as: sitting.
- standing, labeled as: standing.
- lying down on a bed, labeled as: lying down.
- sweeping the floor, labeled as: dynamic standing.
- walking, labeled as: walking.
- walking at 2 different paces on a treadmill, both labeled as: walking.
• cycling on a stationary bicycle at 2 different paces, both labeled: as cycling.

In total this data set contained 2 hours and 23 minutes of recorded data, or about 2430 training examples. The method used to label and record the activities were they same as those used for data set 1.

4.2.4 Training and optimization on the three data sets

The total amount of training data exceeds 21 hours and 15 minutes containing 22813 examples, using windowing techniques described in 3.1.1. The biometric data of the different sets of collected data show that the 52 subjects exhibit very varying anthropomorphic data.

The three feature sets explained in 3.1.4 were extracted from this data and for each feature set learning algorithms were optimized and trained.

Optimal amount of hidden neurons

Artificial neural networks trained with 2 to 30 hidden neurons and the respective 5-folded stratified cross-validation, estimated from 10 runs. Visual inspection reveals that no further significant improvement is gained after hidden layer sizes of; 12, 10, and 8 respectively, see figures 4.2,4.3,4.4. Thus those were chosen as the optimal size of the network.

![Figure 4.2. A hidden neuron search for the feature set containing all features. F-scores are computed from a 5 folded cross-validation averaged over 10 runs. The solid line shows average F-score for all categorize (error bars show the standard deviation from the 10 runs). Dots indicate the average performance for each class.](image)

Optimal amount of pruning

The optimal amount of pruning was estimated by using 5-folded stratified cross-validation, estimated from 10 runs. Visual inspection reveals that little differ be-
4.2. TRAINING DATA

Figure 4.3. A hidden neuron search for the feature set containing the machine selected features. F-scores are computed from a 5 folded cross-validation averaged over 10 runs. The solid line shows average F-score for all categories (errorbars show the standard deviation from the 10 runs). Dots indicate the average performance for each class.

Figure 4.4. A hidden neuron search for the feature set containing the manually selected features. F-scores are computed from a 5 folded cross-validation averaged over 10 runs. The solid line shows average F-score for all categories (errorbars show the standard deviation from the 10 runs). Dots indicate the average performance for each class.

tween the result from the three different feature sets, optimal pruning was chosen to be 30 for all feature sets, see figure 4.5 B.2 B.3.

Optimizing C and gamma for the SVM

The two tunable parameters, C and gamma, were optimized automatically with the grid search method explained in section 3.3 for each binary SVM in the multi-labeled SVM ensembles.
CHAPTER 4. EXPERIMENTAL RESULTS

Figure 4.5. Cross-validated performance averaged over 10 runs for different amount of pruning. The feature set included all constructed features. The solid line shows average F-score for all categories (error bars show the standard deviation from the 10 runs). Dots indicate the average performance for each activity class.

4.3 Testing data

10 females and 10 males were recruited through flyers and advertisements in the classified section of a local newspaper. After arriving at the laboratory they were explained the experimental procedure and presented with an informed consent form. Biometric measures of height and weight were taken and the subjects filled out Baecke questionnaires. After the measurements were taken the multi sensor activity classifier IDEEA was fastened with Medipore® [3M, Minnesota, United States] surgical tape to 5 locations: 2 under the soles of the feet, 2 on the thighs and 1 on the upper sternum. After successful calibration of the IDEEA a second sensor, the Tracmor, was placed inside an elastic belt which the subjects wore around their waist.

After checking that the cables would not interfere with daily activities, using extra surgical tape to securely fasten them if such an act seemed appropriate, the subjects were asked to perform a jump, this produces an easily recognizable reading of the accelerometer and was used as a visual guide to synchronize the devices. Subjects were then asked to perform the following list of activities, listed in order of execution:

- Sitting in an office chair, labeled as: sitting, average segment size: 98.6 [s].
- Standing still, labeled as: standing, average segment size: 115.3 [s].
- Lying down on a bed, labeled as: lying down, average segment size: 106.5 [s].
- Sweeping the floor with a broom, labeled as: dynamic standing, average segment size: 96.0 [s].
4.3. TESTING DATA

- Doing dishes over a sink, labeled as: dynamic standing, average segment size: 96.0 [s].
- Walking through a corridor, labeled as: walking, average segment size: 88.5 [s].
- Walking down a staircase, labeled as: walking, average segment size: 12.7 [s].
- Walking up a staircase, labeled as: walking, average segment size: 14.4 [s].
- Riding an elevator, labeled as: standing, average segment size: 30.0 [s].
- Walking outside, labeled as: walking, average segment size: 115.3 [s].
- Biking around the department building, labeled as: biking, average segment size: 124.3 [s].

The start and stop time of each activity performed was noted by the experimenter and this established the ground truth for the laboratory trial. After completing these activities the subjects were given an activity dairy and told that they should note any of the following activities with start and stop time: Motorized transportation e.g. Train, Bus and Car; Bicycling; Sports activities e.g. running and playing basketball. Subjects were also told to perform a jump before undertaking any such activity, this would later help finding the exact time of the activity in the output of the devices, since subjects clocks were not calibrated with the devices. Subjects were also given detailed instructions on how to remove the devices and to perform a jump before taking them off, this last jump would be used to check for time drift between the two devices. When returning the devices the subjects were given a small compensation for their participation. The whole procedure was reviewed and approved by the internal Philips ethics committee.

4.3.1 Study population data

The subjects professions ranged from fitness trainer and fireguard to research assistant and office workers, the subjects work profiles, however, was predominantly sedentary. This can also readily be seen in the calculated Baecke work score which had an average value of $1.72 \pm 0.58$. Though the majority of the subjects held sedentary work positions most of them exercised, although only 2 subjects performed any exercise during the trial. Their Baecke leisure score suggest that they had quite an active leisure time. This can readily be seen in the amount of biking performed by subjects, average 28 min per subject. Anthropomorphic values for the subjects show that the subjects had a representative span of ages and weight for healthy young subjects (age: $30.6 \pm 9.3$, BMI: $23 \pm 2.6$), see table for a full breakdown on the biometric data.
### 4.3.2 Comparing the outputs of the models in daily life with the golden standard.

The golden standard in this study, IDEEA, does not recognize cycling, for which an activity diary was supplemented. This is a simple yet effective solution since cycling activities usually last for more than 10 minutes and is therefore easy to note down and easy to find when post-processing. Dynamic standing is, unfortunately, not that easy to find in post-processing and causes inconvenience for subjects if they are required to note down every activity that might fall in that category. Furthermore, being a rather fuzzy class label there is no one-to-one correspondence with IDEEA-activities, although IDEEA does recognize some activities that logically would be labeled as dynamic standing, others would fall into the group of “standing”. Therefore, it was decided to merge the category of standing with dynamic standing as doing so would create less ambiguity.

### 4.3.3 Performance of laboratory trained on the laboratory collected data

The laboratory session performed by each subject in the beginning of the trial provided both a measure of the reliability of the golden standard as well as that of the developed models. It should be noted that in this trial the IDEEA is tested on activities it does not have a particular output class for, e.g. sweeping the floor and doing dishes. The accuracy was determined during the bouts of activities for which the ground truth was available. The IDEEA accuracy includes these activities but excludes the cycling, when comparing against IDEEA/diary the IDEEA output is used for all activities but cycling, for which the ground truth is used.

The table, 4.1, lists the population average accuracy for the different models in the laboratory trial against the IDEEA and the ground truth, with the best per-

<table>
<thead>
<tr>
<th></th>
<th>men</th>
<th>women</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>age [years]</td>
<td>22 – 48</td>
<td>24 – 51</td>
<td>22 – 51</td>
</tr>
<tr>
<td></td>
<td>(28.2 ± 7.8)</td>
<td>(32.9 ± 10.4)</td>
<td>(30.6 ± 9.3)</td>
</tr>
<tr>
<td>weight [kg]</td>
<td>60 – 92</td>
<td>50 – 74</td>
<td>50 – 92</td>
</tr>
<tr>
<td></td>
<td>(75.8 ± 10.6)</td>
<td>(61.1 ± 6.5)</td>
<td>(68.4 ± 11.4)</td>
</tr>
<tr>
<td>height [m]</td>
<td>1.68 – 1.83</td>
<td>1.56 – 1.8</td>
<td>1.56 – 1.83</td>
</tr>
<tr>
<td></td>
<td>(1.76 ± 0.06)</td>
<td>(1.68 ± 0.07)</td>
<td>(1.72 ± 0.07)</td>
</tr>
<tr>
<td>BMI [kg/m²]</td>
<td>20.5 – 28</td>
<td>18.8 – 23.4</td>
<td>18.8 – 28.0</td>
</tr>
<tr>
<td></td>
<td>(24.5 ± 2.8)</td>
<td>(21.6 ± 1.4)</td>
<td>(23.0 ± 2.6)</td>
</tr>
<tr>
<td>baecke work score</td>
<td>0.88 – 2.63</td>
<td>1.00 – 2.63</td>
<td>0.88 – 2.63</td>
</tr>
<tr>
<td></td>
<td>(1.70 ± 0.61)</td>
<td>(1.74 ± 0.58)</td>
<td>(1.72 ± 0.58)</td>
</tr>
<tr>
<td>baecke leisure score</td>
<td>2.50 – 4.50</td>
<td>2.50 – 4.25</td>
<td>2.50 – 4.50</td>
</tr>
<tr>
<td></td>
<td>(3.35 ± 0.60)</td>
<td>(3.40 ± 0.60)</td>
<td>(3.38 ± 0.59)</td>
</tr>
</tbody>
</table>
forming models highlighted. An example of the output of the best performing model together with IDEEA classification, the ground truth and accelerometer readings is given for the median performing subject, see figure 4.6.

<table>
<thead>
<tr>
<th>machine learning model</th>
<th>ground truth</th>
<th>IDEEA/diary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>F2</td>
</tr>
<tr>
<td>SVM one-against-one</td>
<td>88.3</td>
<td>83.6</td>
</tr>
<tr>
<td>SVM one-against-all</td>
<td><strong>88.5</strong></td>
<td>84.2</td>
</tr>
<tr>
<td>ANN</td>
<td>85.9</td>
<td><strong>87.1</strong></td>
</tr>
<tr>
<td>ANN bootstrapped</td>
<td>86.1</td>
<td>86.4</td>
</tr>
<tr>
<td>D-tree</td>
<td>79.4</td>
<td>77.1</td>
</tr>
<tr>
<td>D-tree bootstrapped</td>
<td>81.3</td>
<td>80.7</td>
</tr>
<tr>
<td>Plurality voting</td>
<td>86.9</td>
<td>85.0</td>
</tr>
<tr>
<td>IDEEA</td>
<td>94.9</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4.1. Table showing the population average accuracy of the different classifiers in the laboratory trial. The feature sets are in order: F1, all features; F2, machine selected features; F3, manually selected features

Figure 4.6. Supervised trial for the median performing subject of the best performing model, a Support vector machine classifier with one-against-all voting. The Y-axis show class labels 1 sitting, 2 standing, 3 walking, 5 cycling and 6 lying down. For the accelerometer the Y-axis shows the readings in g’s.

4.3.4 Performance of laboratory trained models on daily-life data

Table 4.2, lists the population average accuracy of each model as measured by the golden standard in daily-life. The recorded data spans an entire day so a plot of all the data is not possible, instead the confusion matrix, see table 4.3, for the median performing model is used to illustrates the performance of each individual activity.

A surprising discrepancy between the performance observed in daily life (accuracies of 65 %) and in the laboratory (accuracies close to 85 %) can clearly be seen.
CHAPTER 4. EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>machine learning model</th>
<th>IDEEA/diary</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>F2</td>
<td>F3</td>
<td></td>
</tr>
<tr>
<td>SVM one-against-one</td>
<td>65.6</td>
<td>68.2</td>
<td>65.0</td>
<td></td>
</tr>
<tr>
<td>SVM one-against-all</td>
<td>62.9</td>
<td>59.9</td>
<td>63.1</td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>65.8</td>
<td>65.3</td>
<td>61.1</td>
<td></td>
</tr>
<tr>
<td>ANN bootstrapped</td>
<td>65.5</td>
<td>66.1</td>
<td>61.4</td>
<td></td>
</tr>
<tr>
<td>D-tree</td>
<td>61.0</td>
<td>66.1</td>
<td>62.9</td>
<td></td>
</tr>
<tr>
<td>D-tree bootstrapped</td>
<td>57.5</td>
<td>61.7</td>
<td>66.8</td>
<td></td>
</tr>
<tr>
<td>Plurality voting</td>
<td><strong>67.5</strong></td>
<td><strong>69.5</strong></td>
<td><strong>66.0</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2. Table showing the population average accuracy and standard deviation for the 20 subjects during the daily life trial. The feature sets are in order: F1, all features; F2, machine selected features; F3, manually selected features.

<table>
<thead>
<tr>
<th>IDEEA / Diary</th>
<th>ANF F2 output</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sitting</td>
<td>7.31</td>
<td>34.6</td>
<td>12.4</td>
<td>1.86</td>
<td>0.0826</td>
<td>0.0016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>standing</td>
<td>0.255</td>
<td>4.40</td>
<td>14.8</td>
<td>2.79</td>
<td>1.75</td>
<td>0.0084</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cycling</td>
<td>0</td>
<td>0.0451</td>
<td>0.240</td>
<td>3.22</td>
<td>0.257</td>
<td>0.0021</td>
<td></td>
<td></td>
</tr>
<tr>
<td>walking</td>
<td>0.0056</td>
<td>0.137</td>
<td>0.844</td>
<td>0.933</td>
<td>6.75</td>
<td>0.0307</td>
<td></td>
<td></td>
</tr>
<tr>
<td>running</td>
<td>0</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0772</td>
<td>0.0262</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-score</td>
<td>57.4</td>
<td>72.2</td>
<td>55.9</td>
<td>51.2</td>
<td>76.6</td>
<td>29.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>65.3 ± 7.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3. Confusion matrix for the median performing model, a feed-forward neural network with machine selected features. Values are averaged over all 20 subjects and normalized to sum to 100.

4.4 Analysis

4.4.1 Weighing performance by its distribution in daily-life.

The prevalence of physical activities in daily life was not known before training the algorithms. Therefore the training had little bias with a roughly equal amount of examples for each activity. However, in the daily-life trial the activities were highly skewed towards sedentary activities, see figure 4.7. This affects the estimated performance from the training data and the laboratory data since some activities are easier to detect than others, as seen in figure 4.5. In order to lessen this effect one can weigh the outputs of the algorithms to emulate the distribution of activities found in the daily-life trial.

The weighing was done individually for each subject according to their behavior in the daily-life trial and achieved by multiplying each row of the confusion matrix for that subject. The multiplier used to weigh the row $i$ of the confusion matrix for
4.4. ANALYSIS

Figure 4.7. Percentage of total time spent in different activities for subjects in daily-life recorded with IDEEA and diary.

the laboratory trial, is the ratio of that activity $i$ in the daily-life trial and in the laboratory trial, or:

$$cm_{i,j} = \frac{n_{DL}^i}{n_L^i} \cdot cm_{i,j}$$

where $cm_{i,j}$ is the confusion matrix element where the classifier outputs activity $j$ and the true activity is $i$, $n_{DL}^i$ and $n_L^i$ are the number of occurrences of activity type $i$ in daily life and in the laboratory, respectively. The table 4.4 shows the accuracy of the weighed models.

<table>
<thead>
<tr>
<th>machine learning model</th>
<th>IDEEA/diary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
</tr>
<tr>
<td>SVM one-against-one</td>
<td>80.2</td>
</tr>
<tr>
<td>SVM one-against-all</td>
<td>81.6</td>
</tr>
<tr>
<td>ANN</td>
<td>80.5</td>
</tr>
<tr>
<td>ANN bootstrapped</td>
<td>80.7</td>
</tr>
<tr>
<td>D-tree</td>
<td>75.0</td>
</tr>
<tr>
<td>D-tree bootstrapped</td>
<td>79.9</td>
</tr>
<tr>
<td>Plurality voting</td>
<td><strong>82.0</strong></td>
</tr>
</tbody>
</table>

Table 4.4. Table showing the mean weighed accuracy and standard deviation for the 20 subjects during the laboratory trial. Every subject was weighed according to how they later would act in daily-life. The feature sets are in order: F1, all features; F2, machine selected features; F3, manually selected features.
4.4.2 Training on daily-life data

Early tests showed that training directly on the acquired data with the labels provided by IDEEA/Diary did not generate acceptable results. The skewed distribution made the classifiers emphasize classifying sitting down and standing, this led to accuracies slightly higher than those presented in Table 4.2 but the F-scores for other activities were unacceptable.

An alternative way of classifying the activity data is to weigh the classifications made by the classifiers, as in plurality voting, but adapting the weights depending on performance in daily life. Furthermore, temporal dynamics between classes could be taken into account by considering classifications made by the models in adjacent windows. The simple ANN presented in section 3.6.3 accomplish this within the framework of feed-forward neural networks. Initialization helps the classifier to find a reasonable solution since starting weights make the network work as a plurality voter, the structure also limits the behavior of the ANN, ensuring a reasonable solution. Results from a leave-one-subject out cross-validation are presented in Table 4.5.

<table>
<thead>
<tr>
<th>IDEEA/Diary</th>
<th>Temporal fusion F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lying down</td>
</tr>
<tr>
<td>lying down</td>
<td>1.75</td>
</tr>
<tr>
<td>sitting</td>
<td>0.854</td>
</tr>
<tr>
<td>standing</td>
<td>0.0786</td>
</tr>
<tr>
<td>cycling</td>
<td>0</td>
</tr>
<tr>
<td>walking</td>
<td>0</td>
</tr>
<tr>
<td>running</td>
<td>0</td>
</tr>
</tbody>
</table>

| F-score | 35.8 | 84.1 | 67.7 | 75.0 | 82.7 | 30.3 |
| Accuracy | 77.3 ± 10.3 |

Table 4.5. Aggregated confusion matrix from a leave-one-subject-out cross-validation of the temporal fusion net on daily-life data, inputs to the net are classifiers using all features. Values in the confusion matrix are averaged over all 20 subjects and normalized to sum to 100.
Chapter 5

Discussion

5.1 Results from the laboratory data

The accuracies obtained during the laboratory trial suggest that the algorithms were good at detecting dynamic activities such as walking, running and cycling and one of the postures, lying. However, sitting and standing were more difficult to distinguish from one another, see for example the F-scores of table B.2. This result might not be that surprising as a subject standing still, and a subject sitting still, produce similar readings when measured at the waist. Indeed, from the figure 4.6, the accelerometer reading from sitting and standing still (the two first activities undertaken in the figure) are difficult to distinguish from one another and it is first when the subject is performing activities while standing that the classifier correctly classifies the activity i.e. when the activity is closer to dynamic standing.

The recognition algorithms presented performed well for the other activities on laboratory data, especially in detecting lying down and walking with F-scores varying between 95 - 99, see tables B.1 B.2 B.3. Overall accuracies of the better performing models (88.5% SVM one-against-all, 88.3% SVM one-against-one, 87.9% bagged ANN) are comparable with what has previously been reported[13].

The three feature sets used seemed to have an effect on individual classifiers, sometimes improving model performance with over 5% (5.5% Decision tree). However, the best performing model for each feature set had similar performance (85.9%, 84.6%, 85.3%). This indicates that out of the three proposed feature sets no one can be declared the better set for activity recognition although they could be less or more suited for a particular algorithm.

An observation that could be made is that, although decision trees performed very well in the 5-folded cross-validation, see figure 4.5, recognition rates were lowered during the laboratory trial, see table 4.1. This behavior was not observed for ANNs and SVMs. An explanation for this could be that the Decision trees to a greater extent tune their classifications to subject specific executions, and although it is evaluated on data it has not seen before, the folds created by the cross-validation are very likely to have examples of all subjects and all activities. The decision tree
can therefore construct disconnected areas that classify the same activity, tuned for specific subjects. ANNs and SVMs create smooth boundaries and are therefore not afflicted by this behavior.

5.2 Results from the daily life data

In daily life performance of all the models dropped by a minimum of 10% (Decision tree) and up to 25% (SVM one-vs-all). To some extent this can be attributed to the skewed distribution of classes present in daily-life, which were dominated by the difficult classes (subjects spent approximately 56.6% sitting, see figure 4.7). It is readily seen that the best performing models on the weighed laboratory data perform better than the best models on the unweighed data in the daily life trial.

However, this does not account for all the differences found between results in the laboratory and in daily life, for example the best performing models of each feature set still dropped by 14.5%, 13.2% and 14.4% (Plurality voting) which was statistically significant ($p<0.05$). Furthermore the drops of the weighed F-scores for the activities was statistically significant ($p<0.05$) for these models when compared to the performance in daily life. This difference in detecting daily-life activities compared to laboratory activities indicates that laboratory data is lacking in its ability to represent the activities undertaken in daily life.

The confusion matrices, see tables B.4, B.5 and B.6, seemed to suggest that some particular misclassifications were more frequent than others:

- sitting down was often labeled as lying down.
- sitting and standing were often confused.
- the model labeled the activity as cycling when the subject was sitting or standing.
- poor recognition performance of running.

The first item might be due to the fact that people sometimes reclined in their seats in daily life, whereas in the laboratory they assume a more upright posture. Indeed mean angles seem to suggest that, see figure B.4.

The second item was already detected in the development of the models; discriminating between standing and sitting is difficult with a single waist mounted sensor. The third item seem to suggest that the classifier is overly sensitive for the output cycling, this could be due to the fact that training data lacked examples of sitting and standing containing dynamic movements. Subjects did not perform many bouts of running during the trial, which could account for the poor performance of running, item four.

The method of training a meta-classifier on daily-life data improved accuracies by approximately 10%. It can be seen from the confusion matrix, see table 4.5, that the problem with sitting and lying down is not resolved. The large differences
between daily-life and laboratory data for these two activities made the classifiers weak thus combining the classifiers provided little to no help in improving this classification. Importantly, many of the other misclassifications are not present in this model.

5.3 Summary and Conclusions

This thesis investigated the performance of three machine learning paradigms using features extracted from a waist mounted accelerometer. It was found that while the classifiers provided good performances in predicting the labels of activities performed by subjects in a new laboratory trial, the accuracy in daily life was significantly lower even when the skewed distribution of activities are accounted for (best SVM: \(-13.4 \pm 14.4\%\), p<0.05 best ANN: \(-15.3 \pm 15.9\%\) p<0.05, best Decision tree: \(-20.3 \pm 12.8\%\) p<0.05). From this it was concluded that even when using extensive amounts of laboratory collected data from very different subjects, the accuracy estimation will be overly optimistic if applied in daily-life.

However, by aggregating the laboratory trained models and adapting them to daily-life data, some performance could be regained (\(+9.1 \pm 7.4\%\)). And although this performance \((77 \pm 10.3\%)\) is lower than that which was achieved during the laboratory trials (best model 88.5%) it shows promise for classification using single site devices in daily life.

In terms of the questions asked in the beginning of the thesis it could be concluded that for single-site devices the accuracy achieved compares acceptably to the multi-site device. Although the agreement is not as high as could have been expected from reports using data from laboratory trials[4]. The device still performs well for simple activities such as walking and cycling (F-score of 82.7 and 75). However, there is a clear advantage with multi-site devices in distinguishing the more sedentary activities such as standing (F-score 67.7) and lying down (F-score 35.8) from sitting.

The second question asked i.e. which algorithm is best suited for activity recognition can be answered by examining the performance of the best models during the laboratory trial and the daily life trial. In the weighted laboratory trial the performance of the best model for each paradigm compare quite favorably to each other (81.9 Support vector machine, 81.5 Neural network, 82.6 Decision tree) this is also true in daily-life (68.2 Support vector machine, 65.3 Neural network, 61.7 Decision tree), with support vector machines slightly outperforming the other. Though, support vector machines seem to come off as a slightly stronger program the best performance is gained by combining adjacent time windows and classifiers.

5.4 Limitations

There are several limitations for this study, first the Classifier IDEEA is regarded as the golden standard and even though it has been reported to have high accuracy,
which was confirmed in the laboratory trial, it has not been thoroughly evaluated under daily-life conditions. The estimated performances could be negatively affected by this. This limitation only applies to the absolute values of the performance measures, the drop in classification accuracy between laboratory and daily life still holds although it has been assumed that the source is the developed classifiers not IDEEA.

Secondly, the subjects are all healthy and relatively lean subjects, for other subject groups these result may be different. Furthermore, the subjects were left without any surveillance and so while in the laboratory it could be checked that the sensors never moved the subjects could have turned or manipulated them in daily life thereby lowering the performance.

5.5 Outlook and future work

The results from the trials indicates that algorithms trained and evaluated on laboratory data might perform much worse in a daily-life setting. The unbalanced distribution of activities in daily-life made training on daily-data difficult. However, improvement of activity recognition could be achieved by combining classifiers and including classifications made in adjacent time-windows. Undoubtedly the meta-classifier used in this thesis could be further expanded and future work using waist mounted accelerometers could examine how to best use the temporal information. As one of the weaker points in the meta-classifier was discriminating between sitting and lying down the model might be improved by aggregating the angle as an input to the meta-classifier.

It is found that people in daily-life are predominantly sedentary, with sitting, standing and dynamic standing accounting for 80.6% of all activities. Distinguishing between these activities is inherently difficult for a waist-mounted tri-axial accelerometer. Improvements in recognition might be gained by using single-site devices with more sensors (such as a goniometer or sensitive altimeters) or by using a different body location. Research could also try to investigate recognition of a different group of activities, that might have more distinctive patterns.
Bibliography


Appendix A

Features

This table provides a detailed description of every feature calculated from the window. Notational legend:

\(x_i, y_i, z_i\) the \(i\):th frame of the acceleration data in the X-, Y-, Z-axis. If just \(x\) is used the identical feature is also calculated for the Y and Z components.

\(n\) the length of the window, 128 frames.

\(j\) The imaginary unit, \(\sqrt{-1}\).

LP A lowpass filter for removal of high-frequency noise, explained in section 3.1.2.

BP A bandpass filter removing the DC component as well as high-frequency noise, explained in section 3.1.2.

winsor replacing all data less/greater than the 5:th/95:th percentile with the value of the 5:th/95:th percentile.

hamming A hamming filter employed to remove spectral leakage, explained in section 3.1.2.

\(F(x)\) The amplitude of the frequencies, a vector with the following elements:

\[
\left| \sum_{i} x_i e^{-\frac{2\pi j k}{n}} \right|^2_{k=1,\ldots,n/2}
\]
## APPENDIX A. FEATURES

<table>
<thead>
<tr>
<th>abbreviation</th>
<th>filter</th>
<th>formula</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean()</td>
<td>LP+winsor</td>
<td>[\frac{1}{n} \sum_{i} x_i]</td>
<td>Mean value of the winsored acceleration data.</td>
</tr>
<tr>
<td>std(*)</td>
<td>LP+winsor</td>
<td>[\frac{1}{n-1} \sum_{i} (x_i - \text{mean}(x))^2]</td>
<td>Standard deviation of the winsored acceleration data.</td>
</tr>
<tr>
<td>mangle(*)</td>
<td>LP+winsor</td>
<td>[\text{mean}\left(\text{atan}\left(\frac{x_i}{\sqrt{y_i^2 + z_i^2}}\right)\right)]</td>
<td>Mean angle between the Z-Y plane and the acceleration vector.</td>
</tr>
<tr>
<td>sangle(*)</td>
<td>LP+winsor</td>
<td>[\text{std}\left(\text{atan}\left(\frac{x_i}{\sqrt{y_i^2 + z_i^2}}\right)\right)]</td>
<td>Standard deviation of the angle.</td>
</tr>
<tr>
<td>range()</td>
<td>LP+winsor</td>
<td>[</td>
<td>\max_{i}(x_i) - \min_{i}(x_i)</td>
</tr>
<tr>
<td>mless(*)</td>
<td>LP</td>
<td>[\max(l</td>
<td>x_s-x_{s'} \leq 1, \forall s, s' \in [i, i+l])]</td>
</tr>
<tr>
<td>skew()</td>
<td>LP</td>
<td>[\frac{\sqrt{n} \sum_{i}^n (x_i - \text{mean}(x))^3}{(\sum_{i}^n (x_i - \text{mean}(x))^2)^{3/2}}]</td>
<td>The skewness of the acceleration data.</td>
</tr>
<tr>
<td>kurt()</td>
<td>LP</td>
<td>[\frac{n \sum_{i}^n (x_i - \text{mean}(x))^4}{(\sum_{i}^n (x_i - \text{mean}(x))^2)^2} - 3]</td>
<td>The excess kurtosis of the acceleration data.</td>
</tr>
<tr>
<td>Function</td>
<td>Domain</td>
<td>Formula</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>corr()</td>
<td>LP</td>
<td>[\sum_{i} x_i y_i - n \cdot mean(x_i)mean(y_i) ] [\frac{(n-1)std(x_i)std(y_i)}{(n-1)}]</td>
<td>The correlation between x-axis and y-axis. This feature is also computed for y against z and x against z.</td>
</tr>
<tr>
<td>meanbp()</td>
<td>BP</td>
<td>mean(x)</td>
<td>Mean of the bandpassed signal.</td>
</tr>
<tr>
<td>binFFT()</td>
<td>BP+hamming</td>
<td>[\sum_{i=s}^{s+7} F(x)_s] for s = 1, 9, 17, 25, 33</td>
<td>Energy of frequency subbands with a width of 1.25 Hz.</td>
</tr>
<tr>
<td>binlFFT()</td>
<td>BP+hamming</td>
<td>[\sum_{i=s}^{s+7} \log(F(x)_s+0.05)] for s = 1, ...</td>
<td>A modified logarithm of the energy in the subbands</td>
</tr>
<tr>
<td>energy()</td>
<td>BP+hamming</td>
<td>[\sum_{k} F(x)_k]</td>
<td>the entropy of the amplitude spectra.</td>
</tr>
<tr>
<td>entropy()</td>
<td>BP+hamming</td>
<td>[-\sum_{k} F(x)_k \log(F(x)_k)]</td>
<td>the entropy of the amplitude spectra.</td>
</tr>
<tr>
<td>npeak()</td>
<td>BP+hamming</td>
<td>[#F(x)_k \geq F(x)_s \forall s \in [k-7, k+7]]</td>
<td>number of peaks in the fourier spectrum evaluated in a neighbourhood of 7 frames or (\sim 1.1) Hz.</td>
</tr>
<tr>
<td>apeak()</td>
<td>BP+hamming</td>
<td>[\arg \max_{k} (F(x)_k \geq F(x)_s \forall s \in [...])]</td>
<td>The amplitude of the most prominent peak evaluated in a neighbourhood of 7 frames or (\sim 1.1) Hz.</td>
</tr>
<tr>
<td>Function</td>
<td>Windowing</td>
<td>Feature Description</td>
<td>Calculation</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>bpeak()</code></td>
<td>BP+hamming</td>
<td>The amplitude of the second most prominent peak evaluated in a neighbourhood of 7 frames or $\sim 1.1$ Hz.</td>
<td>$\arg \max_k (F(x)_k \geq F(x)_s)$ for $s = 1, \ldots$</td>
</tr>
<tr>
<td><code>corrFFT()</code></td>
<td>BP+hamming</td>
<td>Cross spectral density transform, also performed for x against z and y against z.</td>
<td>$\sum_{i=s}^{s+7} F(x \ast y)_s$ for $s = 1, \ldots$</td>
</tr>
</tbody>
</table>
Appendix B

Figures

Figure B.1. A 128 frame window applied to a signal with: two frequency components at 1.25 Hz and 1.5 Hz, and one at 0.75 Hz. The first two signals have a magnitude of 0 dB and the third, at 0.75 Hz, has an amplitude of -14 dB. Broadband noise is present at all frequencies at -37.5 dB. The strength of the hamming window is its ability to discern the frequency at 0.75 Hz but because of its broader sidelobe-width it can not always distinguish the frequencies at 1.5 and 1.25 Hz.
Figure B.2. Cross-validated performance averaged over 10 runs for different amount of pruning. The feature set was automatically selected. The solid line shows average F-score for all categories (error bars show the standard deviation from the 10 runs). Dots indicate the average performance for each activity class.

Figure B.3. Cross-validated performance averaged over 10 runs for different amount of pruning. The feature set included the manually selected features. The solid line shows average F-score for all categories (error bars show the standard deviation from the 10 runs). Dots indicate the average performance for each activity class.
<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Best performing model in laboratory trial</th>
<th>Median performing model in laboratory trial</th>
<th>Worst performing model in laboratory trial</th>
</tr>
</thead>
<tbody>
<tr>
<td>lying down</td>
<td>11.1  0.0407 0 0.0318 0.0234 0.0314</td>
<td>11.2  0 0.0318 0 0.0234 0</td>
<td>11.2  0 0 0.0234 0 0</td>
</tr>
<tr>
<td>sitting</td>
<td>0    9.83 1.33 0.0225 0 0</td>
<td>0    9.88 1.32 0 0 0</td>
<td>0    8.32 2.75 0.122 0 0</td>
</tr>
<tr>
<td>standing</td>
<td>0.0509 5.68 28.7 1.60 0.481 0</td>
<td>0.0509 7.04 26.6 2.39 0.513 0</td>
<td>0.090 12.9 21.2 1.84 0.394 0.0635</td>
</tr>
<tr>
<td>cycling</td>
<td>0 0 9.45 13.1 0.506 0</td>
<td>0 0 9.94 12.9 0.707 0</td>
<td>0 0 2.28 11.3 1.03 0</td>
</tr>
<tr>
<td>walking</td>
<td>0 0 0.0071 0.286 0.265 25.9 0.0216</td>
<td>0 0 0.120 0.402 25.9 0.0481</td>
<td>0 0 0.464 0.518 25.32 0.0307</td>
</tr>
<tr>
<td>running</td>
<td>0 0 0 0 0 0</td>
<td>0 0 0 0 0 0</td>
<td>0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**F-score**

- **Table B.1.** Confusion matrix for the best performing model on the unweighed laboratory data, a one-against-all support vector machine classifier using all features. Values are averaged over all 20 subjects and normalized to sum to 100.

- **Table B.2.** Confusion matrix for the median performing model on the unweighed laboratory data, a bootstrapped feed-forward neural network using all features. Values are averaged over all 20 subjects and normalized to sum to 100.

- **Table B.3.** Confusion matrix for the worst performing model on the unweighed laboratory data, a feed-forward neural network with machine selected features. Values are averaged over all 20 subjects and normalized to sum to 100.
### Table B.4

Confusion matrix for the best performing model on the daily life data, a plurality voting classifier using outputs from the classifiers using machine selected features. Values are averaged over all 20 subjects and normalized to sum to 100.

<table>
<thead>
<tr>
<th></th>
<th>lying down</th>
<th>sitting</th>
<th>standing</th>
<th>cycling</th>
<th>walking</th>
<th>running</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDEEA/Diary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lying down</td>
<td>6.59</td>
<td>0.438</td>
<td>0.0557</td>
<td>0.0033</td>
<td>0.0023</td>
<td>0</td>
</tr>
<tr>
<td>sitting</td>
<td>7.95</td>
<td>39.0</td>
<td>8.03</td>
<td>1.24</td>
<td>0.0908</td>
<td>0.0001</td>
</tr>
<tr>
<td>standing</td>
<td>0.215</td>
<td>6.60</td>
<td>13.7</td>
<td>1.73</td>
<td>1.82</td>
<td>0.0131</td>
</tr>
<tr>
<td>cycling</td>
<td>0.0014</td>
<td>0.0814</td>
<td>0.2395</td>
<td>3.14</td>
<td>0.296</td>
<td>0.0025</td>
</tr>
<tr>
<td>running</td>
<td>0.0019</td>
<td>0.1724</td>
<td>0.101</td>
<td>0.526</td>
<td>6.96</td>
<td>0.0247</td>
</tr>
<tr>
<td>F-score</td>
<td>60.3</td>
<td>76.0</td>
<td>58.1</td>
<td>60.4</td>
<td>77.6</td>
<td>30.35</td>
</tr>
</tbody>
</table>

### Table B.5

Confusion matrix for the median performing model on the daily life data, a feed-forward neural network using machine selected features. Values are averaged over all 20 subjects and normalized to sum to 100.

<table>
<thead>
<tr>
<th></th>
<th>lying down</th>
<th>sitting</th>
<th>standing</th>
<th>cycling</th>
<th>walking</th>
<th>running</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDEEA / Diary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lying down</td>
<td>5.9</td>
<td>0.457</td>
<td>0.717</td>
<td>0.0119</td>
<td>0.0022</td>
<td>0.0049</td>
</tr>
<tr>
<td>sitting</td>
<td>7.31</td>
<td>34.6</td>
<td>12.4</td>
<td>1.86</td>
<td>0.0826</td>
<td>0.0016</td>
</tr>
<tr>
<td>standing</td>
<td>0.255</td>
<td>4.40</td>
<td>14.8</td>
<td>2.79</td>
<td>1.75</td>
<td>0.0084</td>
</tr>
<tr>
<td>cycling</td>
<td>0</td>
<td>0.0451</td>
<td>0.240</td>
<td>3.22</td>
<td>0.257</td>
<td>0.0021</td>
</tr>
<tr>
<td>walking</td>
<td>0.0056</td>
<td>0.137</td>
<td>0.844</td>
<td>0.933</td>
<td>6.75</td>
<td>0.0307</td>
</tr>
<tr>
<td>running</td>
<td>0</td>
<td>0.0001</td>
<td>0.0002</td>
<td>0.0009</td>
<td>0.0772</td>
<td>0.0262</td>
</tr>
<tr>
<td>F-score</td>
<td>57.4</td>
<td>72.2</td>
<td>55.9</td>
<td>51.2</td>
<td>76.6</td>
<td>29.4</td>
</tr>
</tbody>
</table>

### Table B.6

Confusion matrix for the worst performing model on the daily life data, a bootstrapped decision tree classifier using all selected features. Values are averaged over all 20 subjects and normalized to sum to 100.

<table>
<thead>
<tr>
<th></th>
<th>lying down</th>
<th>sitting</th>
<th>standing</th>
<th>cycling</th>
<th>walking</th>
<th>running</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDEEA/Diary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lying down</td>
<td>6.80</td>
<td>0.190</td>
<td>0.0940</td>
<td>0.0089</td>
<td>0.0026</td>
<td>0</td>
</tr>
<tr>
<td>sitting</td>
<td>9.21</td>
<td>28.7</td>
<td>16.2</td>
<td>2.05</td>
<td>0.135</td>
<td>0</td>
</tr>
<tr>
<td>standing</td>
<td>0.712</td>
<td>7.53</td>
<td>11.9</td>
<td>2.10</td>
<td>1.78</td>
<td>0.0090</td>
</tr>
<tr>
<td>cycling</td>
<td>0.241</td>
<td>0.0998</td>
<td>0.415</td>
<td>2.83</td>
<td>0.390</td>
<td>0.0014</td>
</tr>
<tr>
<td>walking</td>
<td>0.0235</td>
<td>0.164</td>
<td>1.19</td>
<td>0.496</td>
<td>6.81</td>
<td>0.0163</td>
</tr>
<tr>
<td>running</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0017</td>
<td>0.0001</td>
<td>0.0777</td>
<td>0.0249</td>
</tr>
<tr>
<td>F-score</td>
<td>57.0</td>
<td>61.8</td>
<td>44.3</td>
<td>50.4</td>
<td>76.2</td>
<td>31.9</td>
</tr>
</tbody>
</table>

Table B.4, B.5, B.6 provide the performance metrics for different models trained on the daily life trial data, including F-scores and confusion matrices.
Figure B.4. A histogram showing the mean angle value for the windows identified as sitting and lying down by IDEEA in the daily-life trial, see 3.1.4 for angle definition. The Y-axis indicate the frequency of occurrence normalized so that the sum of all bars equals one i.e. the sum of the bars of one activity will indicate the ratio between the occurrences of the two activities.

Figure B.5. A picture of the TRACMOR next to a standard 1 euro coin. Noticeable in the picture is the usb memory card slot and the USB connection.
Appendix C

Selected code

function [long1] = motionless(Data,tolerance)
datalength = length(Data);
long1 = 1;
for i=1:datalength-1
    lowi = i;
    while((abs(Data(i)-Data(lowi))<tolerance)&&(lowi>1))
        lowi = lowi-1;
    end
    if(long1<i-lowi)
        long1 = i - lowi;
    end
end