A COMPARATIVE STUDY ON HUMAN ACTIVITY CLASSIFICATION WITH MINIATURE INERTIAL AND MAGNETIC SENSORS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Murat Cihan Yüksek

August 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Billur Barshan (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Çiğdem Gündüz Demir

Approved for the Graduate School of Engineering and Sciences:

Prof. Dr. Levent Onural
Director of Graduate School of Engineering and Sciences

ABSTRACT

A COMPARATIVE STUDY ON HUMAN ACTIVITY CLASSIFICATION WITH MINIATURE INERTIAL AND MAGNETIC SENSORS

Murat Cihan Yüksek
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Billur Barshan
August 2011

This study provides a comparative assessment on the different techniques of classifying human activities that are performed using body-worn miniature inertial and magnetic sensors. The classification techniques compared in this study are: naive Bayesian (NB) classifier, artificial neural networks (ANNs), dissimilarity-based classifier (DBC), various decision-tree methods, Gaussian mixture model (GMM), and support vector machines (SVM). The algorithms for these techniques are provided on two commonly used open source environments: Waikato environment for knowledge analysis (WEKA), a Java-based software; and pattern recognition toolbox (PRTools), a MATLAB toolbox. Human activities are classified using five sensor units worn on the chest, the arms, and the legs. Each sensor unit comprises a tri-axial gyroscope, a tri-axial accelerometer, and a tri-axial magnetometer. A feature set extracted from the raw sensor data using principal component analysis (PCA) is used in the classification process. Three different cross-validation techniques are employed to validate the classifiers. A performance comparison of the classification techniques is provided in terms of their correct differentiation rates, confusion matrices, and computational cost.

The methods that result in the highest correct differentiation rates are found to be ANN (99.2%), SVM (99.2%), and GMM (99.1%). The magnetometer is the best type of sensor to be used in classification whereas gyroscope is the least useful. Considering the locations of the sensor units on body, the sensors worn on the legs seem to provide the most valuable information.

Keywords: inertial sensors, gyroscope, accelerometer, magnetometer, activity recognition and classification, feature extraction and reduction, cross validation, Bayesian decision making, artificial neural networks, support vector machines, decision trees, dissimilarity-based classifier, Gaussian mixture model, WEKA, PRTools.

ÖZET

MİNYATÜR EYLEMSİZLİK DUYUCULARI VE MANYETOMETRELER İLE İNSAN AKTİVİTELERİNİN SINIFLANDIRILMASI ÜZERİNE KARŞILAŞTIRMALI BİR ÇALIŞMA

Murat Cihan Yüksek
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Billur Barshan
Ağustos 2011

Bu çalışmada insan vücuduna takılan minyatür eylemsizlik duyucuları ve manyetometreler kullanılarak çeşitli aktiviteler örüntü tanıma yöntemleriyle ayırdedilmiş ve karşılaştırmalı bir çalışmanın sonuçları sunulmuştur. Ayırdetme işlemi için basit Bayesçi (BB) yöntem, yapay sinir ağları (YSA), benzeşmezlik-tabanlı sınıflandırıcı (BTS), çeşitli karar ağacı (KA) yöntemleri, Gauss karışım modeli (GKM) ve destek vektör makinaları (DVM) kullanılmıştır. Kullanılan yöntemlerin algoritmaları, açık kaynak Java tabanlı bir uygulama olan Waikato environment for knowledge analysis (WEKA) ile MATLAB araç kutusu olan pattern recognition toolbox (PRTools) yazılımlarından sağlanmıştır. Aktiviteler gövdeye, kollara ve bacaklara takılan beş duyucu ünitesinden gelen verilerin işlenmesiyle ayırdedilmiştir. Her ünite, her biri üç-eksenli olmak üzere birer ivmeölçer, dönüölçer ve manyetometre içermektedir. Sınıflandırma için ham duyucu verisinden asal bileşenler analizi ile elde edilen öznitelikler kullanılmıştır. Sınıflandırıcılar üç farklı çapraz sağlama yöntemi ile sınanmıştır. Sınıflandırma yöntemlerinin başarımları, başarı oranları, hata matrisleri ve işlem yüklerine göre

karşılaştırılmıştır. Çalışmanın sonuçlarına göre, en iyi ilk üç başarı oranı sırasıyla YSA (%99.2), DVM (%99.2) ve GKM (%99.1) yöntemleri ile elde edilmiştir. Ayırdetme işleminde kullanılabilecek en iyi duyucu tipinin manyetometre, en başarısızının ise dönüölçer olduğu ortaya çıkmıştır. Duyucu ünitelerinin vücut üzerindeki yerleri karşılaştırıldığında ise, bacaklara takılan ünitelerin en değerli bilgileri sağladığı görülmüştür.

Anahtar Kelimeler: eylemsizlik duyucuları, dönüölçer, ivmeölçer, manyetometre, insan aktivitelerinin tanınması ve ayırdedilmesi, öznitelik çıkarma, çapraz sağlama, Bayesçi karar verme, yapay sinir ağları, destek vektör makinaları, karar ağaçları, benzeşmezlik-tabanlı sınıflandırıcı, Gauss karışım modeli, WEKA, PRTools.

ACKNOWLEDGMENTS

I would like to thank everyone who contributed to this thesis. First of all, I wish to express my sincere gratitude to my thesis supervisor Prof. Dr. Billur Barshan for her supervision, guidance, suggestions, and encouragement throughout the development of this thesis.

I would like to express my special thanks and gratitude to Prof. Dr. Enis Çetin and Assist. Prof. Dr. Çiğdem Gündüz Demir for showing keen interest in the subject matter and accepting to read and review the thesis.

I would like to express my appreciation to Kerem Altun, Mustafa Akın Sefünç, and Onur Akın for their contributions to this thesis.

I would also like to thank Scientific and Technological Research Council of Turkey (TÜBİTAK) for financially supporting this work.

It is a great pleasure to express my special thanks to my mother Hülya, father Deniz, and sister Ayşen for their endless love, support, patience, and tolerance.

Contents

List of Figures

List of Tables

List of Acronyms

BDM	Bayesian decision making
NB	naive Bayesian
ANN	artificial neural network
SVM	support vector machines
DBC	dissimilarity-based classifier
NB-T	naive Bayes trees
J48-T	J48 trees
RF-T	random forest
GMM	Gaussian mixture model
GMM$_1$	Gaussian mixture model with one component
GMM$_2$	Gaussian mixture model with two components
GMM$_3$	Gaussian mixture model with three components
GMM$_4$	Gaussian mixture model with four components
EM	Expectation-Maximization
QP	quadratic programming
SMO	sequential minimal optimization
RRSS	repeated random sub-sampling
L1O	leave-one-out
PCA	principal component analysis
FLDA	Fisher linear discriminant analysis
DFT	discrete Fourier transform
WEKA	Waikato environment for knowledge analysis
PRTools	pattern recognition toolbox
MEMS	micro electro-mechanical systems

Chapter 1

Introduction

Inertial sensors are self-contained, nonradiating, nonjammable, dead-reckoning devices that provide dynamic motion information through direct measurements. Gyroscopes provide angular rate information around an axis of sensitivity, whereas accelerometers provide linear or angular velocity rate information.

For several decades, inertial sensors have been used for navigation of aircraft [2, 3], ships, land vehicles, and robots [4, 5, 6], for state estimation and dynamic modeling of legged robots [7, 8], for shock and vibration analysis in the automotive industry, and in telesurgery [9, 10]. Recently, the size, weight, and cost of commercially available inertial sensors have decreased considerably with the rapid development of micro electro-mechanical systems (MEMS) [11]. Some of these devices are sensitive around a single axis; others are multi-axial (usually two- or three-axial). The availability of such MEMS sensors has opened up new possibilities for the use of inertial sensors, one of them being human activity monitoring, recognition, and classification through body-worn sensors [12, 13, 14, 15, 16]. This in turn has a broad range of potential applications in biomechanics [15, 17], ergonomics [18], remote monitoring of the physically or mentally disabled, the elderly, and children [19], detecting and

classifying falls [20, 21, 22], medical diagnosis and treatment [23], home-based rehabilitation and physical therapy [24], sports science [25], ballet and other forms of dance [26], animation and film making, computer games [27, 28], professional simulators, virtual reality, and stabilization of equipment through motion compensation.

Earlier studies in activity recognition employ vision-based systems with single or multiple video cameras, and this remains to be the most common approach to date [29, 30, 31, 32, 33]. For example, although the gesture recognition problem has been well studied in computer vision [34], much less research has been done in this area with body-worn inertial sensors [35, 36]. The use of camera systems may be acceptable and practical when activities are confined to a limited area such as certain parts of a house or office environment and when the environment is well lit. However, when the activity involves going from place to place, camera systems are much less convenient. Furthermore, camera systems interfere considerably with privacy, may supply additional, unneeded information, and cause the subjects to act unnaturally.

Miniature inertial sensors can be flexibly used inside or behind objects without occlusion effects. This is a major advantage over visual motion-capture systems that require a free line of sight. When a single camera is used, the 3-D scene is projected onto a 2-D one, with significant information loss. Points of interest are frequently pre-identified by placing special, visible markers such as light-emitting diodes (LEDs) on the human body. Occlusion or shadowing of points of interest (by human body parts or objects in the surroundings) is circumvented by positioning multiple camera systems in the environment and using several 2-D projections to reconstruct the 3-D scene. This requires each camera to be separately calibrated. Another major disadvantage of using camera systems is that the cost of processing and storing images and video recordings is much higher than those of 1-D signals. 1-D signals acquired from multiple

axes of inertial sensors can directly provide the required information in 3-D. Unlike high-end commercial inertial sensors that are calibrated by the manufacturer, in low-cost applications that utilize these devices, calibration is still a necessary procedure. Accelerometer-based systems are more commonly adopted than gyroscopes because accelerometers are easily calibrated by gravity, whereas gyroscope calibration requires an accurate variable-speed turntable and is more complicated.

The use of camera systems and inertial sensors are two inherently different approaches that are by no means exclusive and can be used in a complementary fashion in many situations. In a number of studies, video cameras are used only as a reference for comparison with inertial sensor data [37, 38, 39, 40, 41, 42]. In other studies, data from these two sensing modalities are integrated or fused [43, 44]. The fusion of visual and inertial data has attracted considerable attention recently because of its robust performance and potentially wide applications [45, 46]. Fusing the data of inertial sensors and magnetometers is also reported in the literature [40, 47, 48].

Previous work on activity recognition based on body-worn inertial sensors is fragmented, of limited scope, and mostly unsystematic in nature. Due to the lack of a common ground among different researchers, results published so far are difficult to compare, synthesize, and build upon in a manner that allows broad conclusions to be reached. A unified and systematic treatment of the subject is desirable; theoretical models need to be developed that will enable studies designed such that the obtained results can be synthesized into a larger whole.

Most previous studies distinguish between sitting, lying, and standing [19, 37, 38, 39, 42, 49, 50, 51, 52], as these postures are relatively easy to detect using the static component of acceleration. Distinguishing between walking, and ascending and descending stairs has also been accomplished [49, 50, 52], although not as successfully as detecting postures. The signal processing and motion detection

techniques employed, and the configuration, number, and type of sensors differ widely among the studies, from using a single accelerometer [19, 53, 54, 55] to as many as 12 [56] on different parts of the body. Although gyroscopes can provide valuable rotational information in 3-D, in most studies, accelerometers are preferred to gyroscopes because of the ease of calibration. To the best of our knowledge, guidance on finding a suitable configuration, number, and type of sensors does not exist [49]. Usually, some configuration and some modality of sensors are chosen without strong justification, and empirical results are presented. Processing the acquired signals is also often done ad hoc and with relatively unsophisticated techniques. A summary of the sensor configuration, classified activities, the subjects, classification techniques with the corresponding maximum correct differentiation rate reported in earlier studies can be found in Table 1.1. This

Ref.	sensors		activity		subjects		classification technique		maximum correct differentiation rate (%)
	number	type	number	type	male	female	number	best method	
[16]	2	gyro	8	mot	1	N/A	7	BDM	98.2
[18]	23	acc, mag, GPS, other	7	pos, mot	13	3	3	custom decision tree	97.0
[19]	1	acc	12	pos, mot, trans	19	7	1	hierarchical decision tree	97.7
[25]	5	acc, GPS	20	pos, mot	10	2	4	hybrid model classifier	N/A
[39]	2	acc	5	pos, mot	1	4	1	physical activity detection algorithm	89.7
[49]	6	acc	20	pos, mot	13	7	4	C4.5 decision tree	84.3
[54]	1	acc	8	pos, mot	2	4	3	adopted GMM	92.2
[55]	1	acc	19	pos, mot, trans	3	3	1	hierarchical recognizer	97.9
[56]	12	acc	8	pos, mot	1	N/A	1	BDM	N/A
[57]	15	gyro, acc, mag	19	pos, mot	4	4	7	BDM	99.2
[58]	12	video tags	6	pos, mot	3	N/A	8	SVM	N/A

Table 1.1: A summary of earlier studies on activity recognition. The information provided from leftmost to rightmost column are: the reference number, number and type of sensors [gyroscope (gyro), accelerometer (acc), magnetometer (mag), global positioning system (GPS), other (other type of sensors)], number of activities classified, basic group of activities [posture (pos), motion (mot), transition (trans)], number of male and female subjects, number of classification methods, the best method, and the correct differentiation rate of the best method.

study is an extension of the earlier work performed by our research group and

reported in [57]. In that work, miniature inertial sensors and magnetometers positioned on different parts of the body are used to classify human activities. The motivation behind investigating activity classification is its potential applications in the many different areas mentioned above. The main contribution of the earlier article is that unlike previous studies, many redundant sensors are used to begin with and a variety of features from the sensor signals are extracted. Then, unsupervised feature transformation technique that allows considerable feature reduction through automatic selection of the most informative features are used. Extensive and systematic comparison between various classification techniques used for human activity recognition based on the same data set is provided. The classification techniques evaluated are least-squares method (LSM), k-nearest neighbor (k-NN), dynamic time warping (DTW), rule-based algorithm (RBA), Bayesian decision making (BDM), support vector machines (SVM), and artificial neural networks (ANNs). The correct differentiation rates, confusion matrices, and computational requirements of the techniques are compared.

In this study, we evaluate the performance of alternative classification techniques on the data set used previously. The classification methodology in terms of feature extraction and reduction and cross-validation techniques are kept the same. In [57], the algorithms compared are implemented by the authors, whereas the algorithms considered in this study are provided in two open source environments in which a wide variety of classification algorithms are available. These environments are Waikato environment for knowledge analysis (WEKA) and pattern recognition toolbox (PRTools). WEKA is a Java based collection of machine learning algorithms for solving data mining problems [59, 60]. PRTools is a MATLAB based toolbox for pattern recognition [61]. WEKA is executable via MATLAB so that MATLAB is used as the master software to manage both environments. The performances of these two environments are compared in terms of the classification performance and execution time of the algorithms employed. The shorter version of this work appears in [62] and [63].

The rest of this thesis is organized as follows: In Chapter 2, classified activities and data acquisition methodology are explained and descriptions of the features used and the feature vectors, and the feature reduction approach are given. In Chapter 3, classification techniques are reviewed. In Chapter 4, experimental results, comparison of the classification techniques, and time considerations are presented. In Chapter 5, some conclusions are drawn, several potential applications of this study are mentioned and future research directions are discussed.

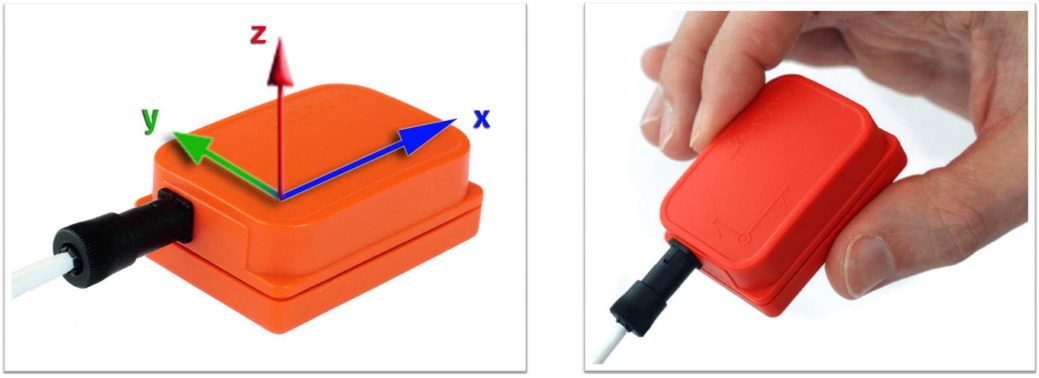
Chapter 2

Experimental Methodology and Feature Extraction

In this chapter, classified activities and data acquisition methodology are explained and descriptions of the features used, the feature vectors, and the feature reduction approach are given.

2.1 Experimental Methodology

The 19 activities that are classified using body-worn miniature inertial sensor units are: sitting (A1), standing (A2), lying on back and on right side (A3 and A4), ascending and descending stairs (A5 and A6), standing in an elevator still (A7) and moving around (A8), walking in a parking lot (A9), walking on a treadmill with a speed of 4 km/h (in flat and 15° inclined positions) (A10 and A11), running on a treadmill with a speed of 8 km/h (A12), exercising on a stepper (A13), exercising on a cross trainer (A14), cycling on an exercise bike in horizontal and vertical positions (A15 and A16), rowing (A17), jumping (A18), and playing basketball (A19).



(a) (b)

Figure 2.1: (a) MTx with sensor-fixed coordinate system overlaid, (b) MTx held in a palm (both parts of the figure are reprinted from [1]).

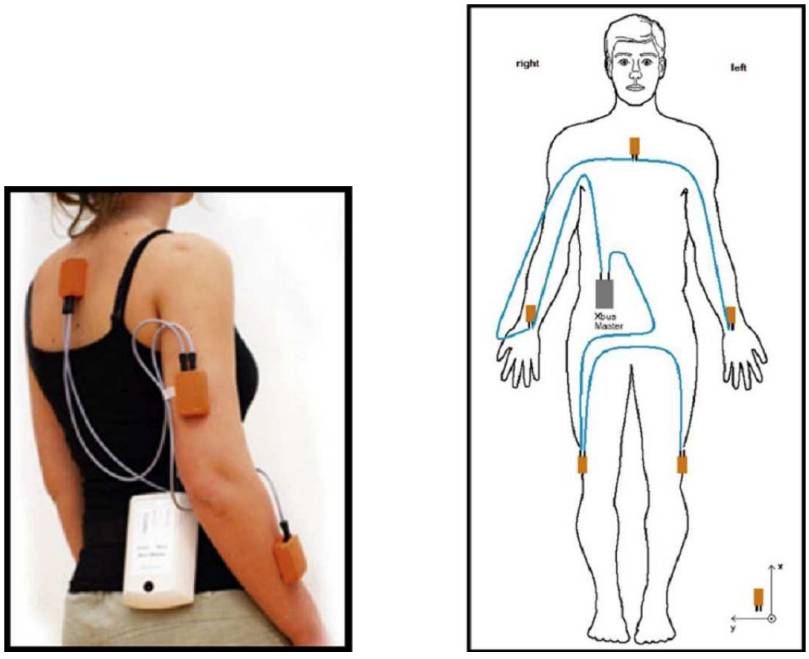
Five MTx 3-DOF orientation trackers (Figure 2.1) are used, manufactured by Xsens Technologies [1]. Each MTx unit has a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer, so the sensor units acquire 3-D acceleration, rate of turn, and the strength of Earth's magnetic field. Each motion tracker is programmed via an interface program called MT Manager to capture the raw or calibrated data with a sampling frequency of up to 512 Hz. Accelerometers of two of the MTx trackers can sense up to $\pm 5g$ and the other three can sense in the range of $\pm 18g$, where $g = 9.80665$ m/s^2 is the standard gravity. All gyroscopes in the MTx unit can sense in the range of $\pm 1200°$/sec angular velocities; magnetometers can sense magnetic fields in the range of $\pm 75\mu$T. We use all three types of sensor data in all three dimensions. The sensors are

(a) (b) (c)

Figure 2.2: Positioning of Xsens sensor modules on the body.

placed on five different places on the subject's body as depicted in Figure 2.2. Since leg motions, in general, may produce larger accelerations, two of the $\pm 18g$ sensor units are placed on the sides of the knees (right side of the right knee and left side of the left knee), the remaining $\pm 18g$ unit is placed on the subject's chest (Figure 2.2(b)), and the two $\pm 5g$ units on the wrists (Figure 2.2(c)).

The five MTx units are connected with 1 m cables to a device called the Xbus Master, which is attached to the subject's belt. The Xbus Master transmits data from the five MTx units to the receiver using a Bluetooth^{TM} connection. The Xbus Master, which is connected to three MTx orientation trackers, can be seen in Figure 2.3(a). The receiver is connected to a laptop computer via a USB port. Two of the five MTx units are directly connected to the Xbus Master and the remaining three units are indirectly connected to the Xbus Master by wires to the other two. Figure 2.3(b) illustrates the connection configuration of the five MTx units and the Xbus Master. Each activity listed above is performed



(a) (b)

Figure 2.3: (a) MTx blocks and Xbus Master (the picture is reprinted from http://www.xsens.com/en/movement-science/xbus-kit), (b) connection diagram of MTx sensor blocks (body part of the figure is from http://www.answers.com/body breadths).

by eight different subjects (four female, four male, between the ages 20 and 30) for 5 min. The subjects are asked to perform the activities in their own style and were not restricted on how the activities should be performed. For this reason, there are inter-subject variations in the speeds and amplitudes of some activities. The activities are performed at the Bilkent University Sports Hall, in the Electrical and Electronics Engineering Building, and in a flat outdoor area on campus. Sensor units are calibrated to acquire data at 25 Hz sampling frequency. The 5-min signals are divided into 5-s segments, from which certain features are extracted. In this way, 480 ($= 60 \times 8$) signal segments are obtained for each activity.

2.2 Feature Extraction and Reduction

After acquiring the signals as described above, we obtain a discrete-time sequence of N_s elements that can be represented as an $N_s \times 1$ vector $\mathbf{s} = [s_1, s_2, \ldots, s_{N_s}]^T$. For the 5-s time windows and the 25-Hz sampling rate, $N_s = 125$. The initial set of features we use before feature reduction are the minimum and maximum values, the mean value, variance, skewness, kurtosis, autocorrelation sequence, and the peaks of the discrete Fourier transform (DFT) of $\mathbf{s}$ with the corresponding

frequencies. These are calculated as follows:

$$\text{mean}(\mathbf{s}) = \mu_{\mathbf{s}} = E\{\mathbf{s}\} = \frac{1}{N_s}\sum_{i=1}^{N_s} s_i$$

$$\text{variance}(\mathbf{s}) = \sigma_{\mathbf{s}}^2 = E\{(\mathbf{s}-\mu_{\mathbf{s}})^2\} = \frac{1}{N_s}\sum_{i=1}^{N_s}(s_i-\mu_{\mathbf{s}})^2$$

$$\text{skewness}(\mathbf{s}) = \frac{E\{(\mathbf{s}-\mu_{\mathbf{s}})^3\}}{\sigma_{\mathbf{s}}^3} = \frac{1}{N_s\sigma_{\mathbf{s}}^3}\sum_{i=1}^{N_s}(s_i-\mu_{\mathbf{s}})^3$$

$$\text{kurtosis}(\mathbf{s}) = \frac{E\{(\mathbf{s}-\mu_{\mathbf{s}})^4\}}{\sigma_{\mathbf{s}}^4} = \frac{1}{N_s\sigma_{\mathbf{s}}^4}\sum_{i=1}^{N_s}(s_i-\mu_{\mathbf{s}})^4$$

$$\text{autocorrelation}: \quad R_{\mathbf{ss}}(\Delta) = \frac{1}{N_s-\Delta}\sum_{i=0}^{N_s-\Delta-1}(s_i-\mu_{\mathbf{s}})\,(s_{i-\Delta}-\mu_{\mathbf{s}})$$

$$\text{where } \Delta = 0, 1, \ldots, N_s-1$$

$$\text{DFT}: \qquad S_{\text{DFT}}(k) = \sum_{i=0}^{N_s-1} s_i\, e^{-\frac{j2\pi ki}{N_s}}$$

$$\text{where } k = 0, 1, \ldots, N_s-1$$

In these equations, s_i is the ith element of the discrete-time sequence $\mathbf{s}$, $E\{\cdot\}$ denotes the expectation operator, $\mu_{\mathbf{s}}$ and $\sigma_{\mathbf{s}}$ are the mean and the standard deviation of $\mathbf{s}$, $R_{\mathbf{ss}}(\Delta)$ is the unbiased autocorrelation sequence of $\mathbf{s}$, and $S_{\text{DFT}}(k)$ is the kth element of the 1-D N_s-point DFT. In calculating the first five features above, it is assumed that the signal segments are the realizations of an ergodic process so that ensemble averages are replaced with time averages. Apart from those listed above, we have also considered using features such as the total energy of the signal, cross-correlation coefficients of two signals, and the discrete cosine transform coefficients of the signal.

Since there are five sensor units (MTx), each with three tri-axial devices, a total of nine signals are recorded from every sensor unit. When a feature such as the mean value of a signal is calculated, 45 (= 9 axes × 5 units) different values are available. These values from the five sensor units are placed in the feature vectors in the order of right arm (RA), left arm (LA), right leg (RL), torso (T), and left leg (LL). For each one of these sensor locations, nine values for each feature are calculated and recorded in the following order: the x, y, z axes'

acceleration, the x, y, z axes' rate of turn, and the x, y, z axes' Earth's magnetic field. In constructing the feature vectors, the above procedure is followed for the minimum and maximum values, the mean, skewness, and kurtosis. Thus, $225\,(= 45 \text{ axes} \times 5 \text{ features})$ elements of the feature vectors are obtained by using the above procedure.

After taking the DFT of each 5-s signal, the maximum five Fourier peaks are selected so that a total of $225\,(= 9 \text{ axes} \times 5 \text{ units} \times 5 \text{ peaks})$ Fourier peaks are obtained for each segment. Each group of 45 peaks is placed in the order of RL, LA, RL, T, and LL, as above. The 225 frequency values that correspond to these Fourier peaks are placed after the Fourier peaks in the same order.

Eleven autocorrelation samples are placed in the feature vectors for each axis of each sensor, following the order given above. Since there are 45 distinct sensor signals, $495\,(= 45 \text{ axes} \times 11 \text{ samples})$ autocorrelation samples are placed in each feature vector. The first sample of the autocorrelation function (the variance) and every fifth sample up to the fiftieth are placed in the feature vectors for each signal.

As a result of the above feature extraction process, a total of $1,170\,(= 225 + 225 + 225 + 495)$ features are obtained for each of the 5-s signal segments so that the dimensions of the resulting feature vectors are $1,170 \times 1$. All features are normalized to the interval $[0, 1]$ so as to be used for classification.

Because the initial set of features was quite large (1,170) and not all features were equally useful in discriminating between the activities, we investigated different feature selection and reduction methods [64]. In this work, we reduced the number of features from 1,170 to 30 through principal component analysis (PCA) [65], which is a transformation that finds the optimal linear combinations of the features, in the sense that they represent the data with the highest variance in a feature subspace, without taking the intra-class and inter-class

variances into consideration separately. The reduced dimension of the feature vectors is determined by observing the eigenvalues of the covariance matrix of the $1,170 \times 1$ feature vectors, sorted in Figure 2.4(a) in descending order. The 30 eigenvectors corresponding to the largest 30 eigenvalues (Figure 2.4(b)) are used to form the transformation matrix, resulting in 30×1 feature vectors. Although the initial set of 1,170 features do have physical meaning, because of the matrix transformation involved, the transformed feature vectors cannot be assigned any physical meaning. Scatter plots of the first five transformed features are given in Figure 2.5 pairwise. As expected, in the first two plots or so (parts (a) and (b) of the figure), the features for different classes are better clustered and more distinct. We assume that after feature reduction, the resulting feature vector is an $N \times 1$ vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$.

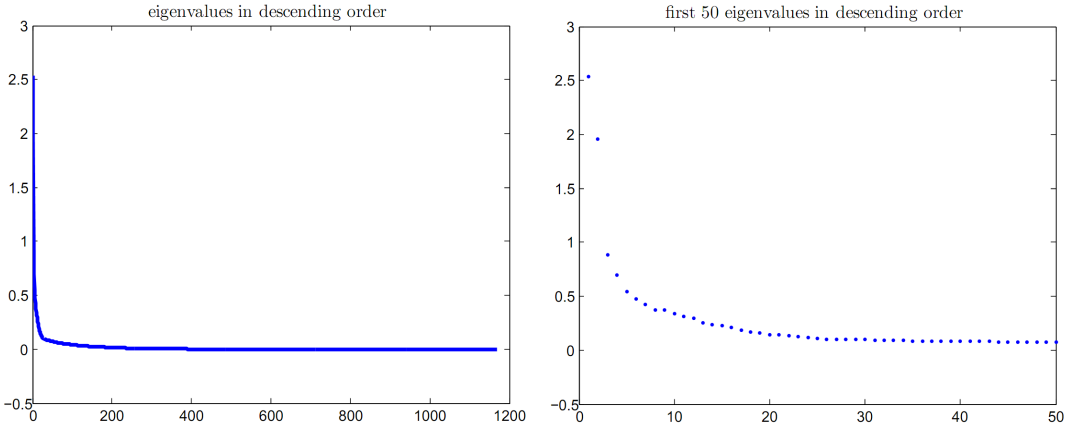


Figure 2.4: (a) All 1,170 eigenvalues, (b) the first 50 eigenvalues of the covariance matrix sorted in descending order.

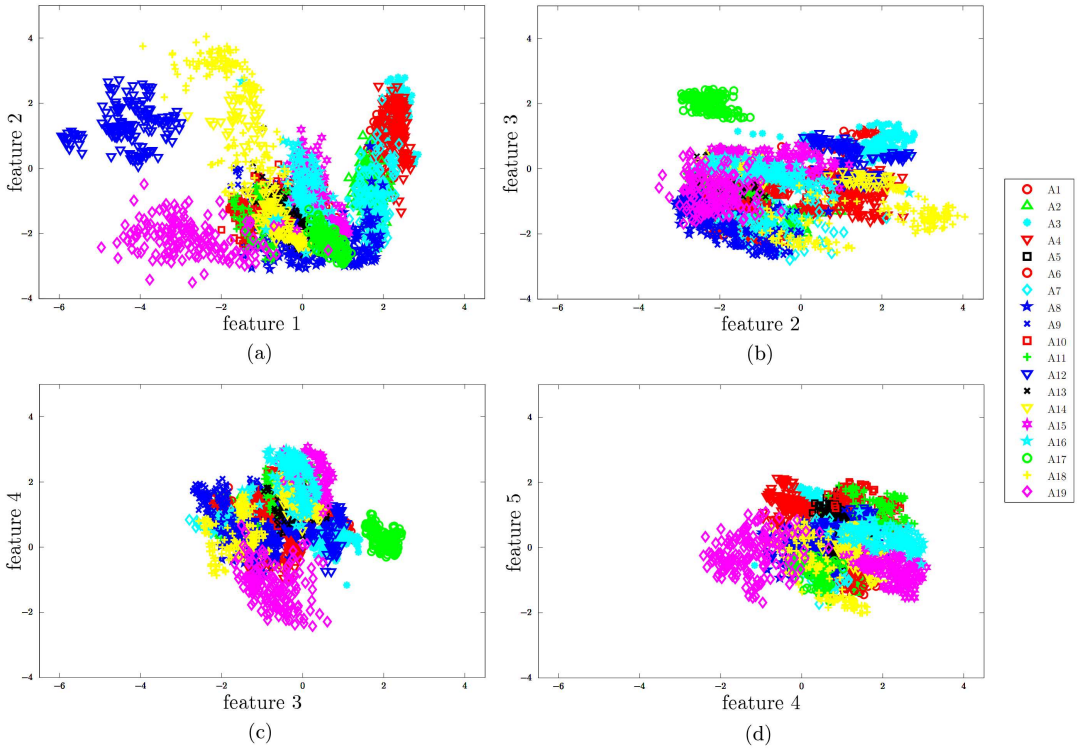


Figure 2.5: Scatter plots of the first five features selected by PCA.

Chapter 3

Classification Techniques

The classification techniques used in this study are briefly reviewed in this chapter. We associate a class w_j with each activity type ($j = 1, 2, \ldots, c$). Every feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ in the set of training patterns $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$ is labeled with corresponding class w_j if it falls in the region Ω_j. A rule that partitions the decision space into regions Ω_j is called a *decision rule*. In our work, each one of these regions corresponds to a different activity type. Boundaries between these regions are called *decision surfaces*. The training set contains a total of $I = I_1 + I_2 + \ldots + I_c$ sample feature vectors where I_j sample feature vectors belong to class w_j. In the training set, the number of feature vectors included in w_j depends on the cross-validation method employed. The *test set* is then used to evaluate the performance of the decision rule.

3.1 Naive Bayesian (NB)

Naive Bayes classifier is based on the Bayes' theorem and calculates the posterior probabilities according to the probabilistic models of each class. In this method, $p(\mathbf{x}_i|w_j)$ denotes the class conditional probability density function given

the class w_j. Probabilistic models for $p(\mathbf{x}_i|w_j)$ are constructed first, using the training data for each w_j. The probability density function is modeled as a normal distribution whose parameters (mean and variance) are estimated by maximum likelihood estimation. A simplifying assumption in the NB method is that the features are independent of each other and model parameters are calculated accordingly. Prior probabilities are taken to be equal and the posterior probabilities are calculated as $p(w_j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|w_j)p(w_j)}{p(\mathbf{x}_i)}$, where $p(\mathbf{x}_i) = \sum_{j=1}^{c} p(\mathbf{x}_i|w_j)p(w_j)$ is the total probability. Classification is made based on maximum a posteriori (MAP) decision rule so that the feature vector is assigned to the class with the highest posterior probability [66].

3.2 Artificial Neural Networks (ANNs)

The theory underlining ANNs is inspired by the working principles of actual neurons in the brain. The main purpose of ANNs is to learn nonlinear mapping parameters along with linear discriminant parameters simultaneously so that highly complex data mining and classification tasks are feasible [67]. A multi-layer ANN consists of an input layer, one or more hidden layers to extract progressively more meaningful features, and a single output layer, each composed of a number of units called neurons. The model of each neuron includes a smooth nonlinearity, called the *activation function*. Due to the presence of distributed nonlinearity and a high degree of connectivity, theoretical analysis of ANNs is difficult. These networks compute the boundaries of decision regions by adjusting their connection weights and biases through the use of training algorithms. The performance of ANNs is affected by the choice of parameters related to the network structure, training algorithm, and input signals, as well as by parameter initialization [68, 69].

In this work, a three-layer artificial neural network (ANN) is used for classifying human activities. The input layer has N neurons, equal to the dimension of the feature vectors (30), the hidden layer has $N + c$ neurons, equal to the sum of the dimension of the feature vectors and the number of classes (49), and the output layer has c neurons, equal to the number of classes (19). For an input feature vector $\mathbf{x} \in \mathbb{R}^N$, the target output is one for the class that the vector belongs to, and zero for all other output neurons. The sigmoid function used as the activation function in the input and output layers is given by $h(x) = (1 + e^{-x})^{-1}$.

The output neurons can take continuous values between zero and one. Fully connected ANNs are trained with the back-propagation algorithm which is the extension of the least mean squares (LMS) method and based on the gradient-descent algorithm [67, 69, 70] by presenting a set of feature vectors to the network. The aim is to minimize the average of the sum of squared errors over all training vectors:

$$\mathcal{E}_{\text{av}}(\mathbf{w}) = \frac{1}{2I} \sum_{i=1}^{I} \sum_{j=1}^{c} [t_{ij} - o_{ij}(\mathbf{w})]^2 \tag{3.1}$$

Here, $\mathbf{w}$ is the weight vector, t_{ij} and o_{ij} are the desired and actual output values for the ith training feature vector and the jth output neuron, and I is the total number of training feature vectors as before. When the entire training set is covered, an *epoch* is completed. The error between the desired and actual outputs is computed at the end of each iteration and these errors are averaged at the end of each epoch (Equation (3.1)). The training process is terminated when a certain precision goal on the average error is reached or if the specified maximum number of epochs (1,000) is exceeded. Precision goal and weight vector initializations are made by the classification toolboxes themselves. A three-layer ANN with learning and momentum constants both set equal to 0.05 is employed.

3.3 Dissimilarity-Based Classifier (DBC)

In DBC, a classifier based on Fisher linear discriminant analysis (FLDA) is developed using the data that are obtained by a dissimilarity mapping of the original feature vectors. The notion of *dissimilarity space* in which objects are characterized by relation to other objects instead of features or models is a recent concept in pattern recognition [71]. In this study, the feature vectors in X are treated as objects and the method is implemented on those feature vectors. It is shown that working on dissimilarity spaces derived from feature vectors yields some interesting results [72].

A dissimilarity mapping is defined as $F(\cdot, R) : X \rightarrow \mathbb{R}^n$ from X to so called *dissimilarity space*. The n-element set R consists of feature vectors that are representative for the problem. This set is called the representation set and it can be any subset of X. In this study, the vectors in R are chosen randomly with $n = 100$ so that a representation set $R = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ is formed. An n-dimensional dissimilarity vector $F(\mathbf{x}, R) = [u(\mathbf{x}, \mathbf{r}_1), \ldots, u(\mathbf{x}, \mathbf{r}_n)]^T$ between the feature vector $\mathbf{x}$ and the set R describes the resulting objects. An Euclidean dissimilarity measure ρ, between $\mathbf{x}$ and $\mathbf{x}'$, is defined in *dissimilarity space* to be used in the test stage of the classification:

$$\rho(\mathbf{x}, \mathbf{x}') = \sum_{\ell=1}^{n} [u(\mathbf{x}, \mathbf{r}_\ell) - u(\mathbf{x}', \mathbf{r}_\ell)] \tag{3.2}$$

As a result, the feature space is mapped onto the n-dimensional *dissimilarity space*. The linear discriminant functions are found using FLDA by minimizing the errors in the least square sense. In FLDA, the criterion function

$$J(\mathbf{W}) = \frac{\mid \mathbf{W^T S_B W} \mid}{\mid \mathbf{W^T S_W W} \mid} \tag{3.3}$$

is to be maximized. In Equation (3.3), $\mathbf{W}$, $\mathbf{S_B}$, and $\mathbf{S_W}$ are $N \times (c-1)$ transformation matrix, *between-class*, and *within-class scatter matrices*, respectively. The operator $\mid \cdot \mid$ denotes the determinant. The *scatter matrices* are expressed

as

$$\mathbf{S_B} = \sum_{j=1}^{c} I_j(\boldsymbol{\mu}_j - \boldsymbol{\mu}_\mathbf{x})(\boldsymbol{\mu}_j - \boldsymbol{\mu}_\mathbf{x})^T \tag{3.4}$$

$$\mathbf{S_W} = \sum_{j=1}^{c} \sum_{\mathbf{x} \in w_j} (\mathbf{x} - \boldsymbol{\mu}_j)(\mathbf{x} - \boldsymbol{\mu}_j)^T \tag{3.5}$$

where $\boldsymbol{\mu}_\mathbf{x} = \frac{1}{I} \sum_{i=1}^{I} \mathbf{x}_i$ and $\boldsymbol{\mu}_j = \frac{1}{I_j} \sum_{\mathbf{x} \in w_j} \mathbf{x}$. As before, I_j denotes the number of feature vectors in the jth class. It can be shown that $J(\mathbf{W})$ is maximized when the columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{S_W}^{-1}\mathbf{S_B}$ having the largest eigenvalues. As a result, $c-1$ classifiers are built to perform the classification in c-dimensional space.

3.4 Decision-Tree Methods

Decision-tree classifiers are non-metric classifiers in which no measure of distance can be found so that they are efficiently adapted to tasks where nominal features appear. Nominal features are non-numeric and descriptive features such as those that specify the color of an object (e.g., green, red, blue, etc.). However, real-valued features can also be used in the classification process. Decision-tree classifiers are fast, comprehensible, and easy to visualize.

Decision-tree induction is based on *divide and conquer* algorithm that recursively breaks down a problem into two or more subproblems until these problems of related type are directly solvable. In decision-tree notion, directly solvable problems indicate the leaf nodes. In most of the decision-tree methods including the ones used here, each node along with the root, is split into two branches considering a single feature according to some criterion. The process continues until a leaf is encountered. The leaf is a node at which the class of a given feature vector is indicated. There are several important aspects of decision-tree induction methods: number of splits at a node, splitting criterion, and stopping criterion. There is another term called *pruning* which reduces the size of the tree

by considering all pairs of neighboring leaf nodes for elimination after a complete tree is built. Pruning prevents overfitting [67, 73].

WEKA is used for the decision-tree classification tests. The correct differentiation rates acquired seems to be robust to changes in classifier parameters during the implementation; therefore, default parameters are used. Pruning is performed on the generated trees which is the only change in the parameter settings. One of the striking drawbacks of WEKA is that some tree methods are not applicable because of the memory restrictions of the software.

3.4.1 Trees Using J48 Algorithm (J48-T)

J48 method implements the C4.5 algorithm for generating a pruned or an unpruned C4.5 decision-tree learner which is an improved version of the ID3 learner. Both ID3 and C4.5 algorithms are developed by Ross Quinlan [74]. ID3 allows only two classes, requires nominal or discrete features, and does not deal with the feature vectors comprising missing and noisy features. C4.5, on the other hand, can be used for classification tasks involving multiple classes and feature vectors with real-valued, missing, and noisy features [75, 76].

J48 builds decision trees from a set of labeled training data using the concept of normalized information gain. This concept is a splitting criterion that is used for selecting the feature that most effectively splits the given set of feature vectors at a tree node. It is desired to define a *rule*, ϑ, at a node for splitting, based on a single feature of a feature vector, $\mathbf{x} = [x_1,x_2,\ldots,x_N]$, such that the selected feature, x_k, will yield the maximum normalized information gain. The rule ϑ determines the structure of the subtree of the node that it belongs to and C4.5 uses three types of *rules* for splitting at a node:

- If x_k is a discrete feature with L outcomes, possible queries that will constitute ϑ are:

    1. "$x_k = ?$,"for all possible L outcomes of x_k.
    2. "$x_k \in G_?$" with $2 \leq l \leq L$ outcomes, where $G = \{G_1,\ldots,G_l\}$ is a partition of the values of x_k. G is determined with a greedy search according to the splitting criterion which is information gain (discussed below).

- If x_k is real-valued, the query becomes:

    3. "$x_k \leq \xi$" with outcomes true or false, where ξ is a constant threshold. Each possible value of x_k is considered to find ξ. If there are d possible values of x_k, $d-1$ possible thresholds are considered between each pair of adjacent values.

The class of a feature vector in X is identified by its information content which is expressed as:

$$B(X) = -\sum_{j=1}^{c} Fr(w_j, X)\log\big(Fr(w_j, X)\big) \tag{3.6}$$

where $Fr(w_j, X)$ denotes the relative frequency of the feature vectors in X that belong to class w_j. Once X is partitioned into subsets X_1, X_2,..., X_Q by ϑ, the information gained is calculated with the following equation:

$$A_B(X, \vartheta) = B(X) - \sum_{q=1}^{Q} \frac{\mid X_q \mid}{\mid X \mid} B(X_q) \tag{3.7}$$

The potential information in a partition X_q can be found using the following expression:

$$A_P(X, \vartheta) = -\sum_{q=1}^{Q} \frac{\mid X_q \mid}{\mid X \mid} \log\left(\frac{\mid X_q \mid}{\mid X \mid}\right) \tag{3.8}$$

The *rule* that maximizes the normalized information gain, $A_N(X, \vartheta) = \frac{A_B(X,\vartheta)}{A_P(X,\vartheta)}$, is chosen along with the feature to be used at a node. If all feature vectors in X_q

belong to the same class, a leaf node is created. If none of the features provide any information gain, a decision node higher up the tree is created. If neither one of the previous cases occur, a child node is created.

3.4.2 Naive Bayes Trees (NB-T)

Naive Bayes trees are hybrid classifiers that combine the principles governing the NB classifier and decision-tree classifiers. The hybrid algorithm is similar to the classical recursive decision-tree partitioning schemes, except that the leaf nodes created are NB classifiers instead of nodes predicting a single class. The main drawback of NB method is that if the assumptions regarding the independence of features fail, performance cannot be improved by increasing the size of the dataset [73].

Given a feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ for training, the threshold, ξ, is calculated for real-valued features using the normalized information gain concept defined in Section 3.4.1. In addition, the utility function, $U(x_k)$, is used to find the utility of a split on x_k by discretizing the feature vectors and computing the five-fold cross-validation accuracy estimate of using NB at that node. The utility of a split is the weighted sum of the utility of the nodes, where the weight given to a node is proportional to the number of feature vectors that go down to that node. The feature with the maximum utility such that

$$k_{\max} = \arg\max_{k} U(x_k) \tag{3.9}$$

is determined. If $U(x_{k_{\max}})$ is not *significantly* better than the utility of the current node, a NB classifier is created for the current node. Here, the term *significance* implies that the relative reduction in error is greater than 5% and there are at least 30 feature vectors in the node. If *significance* is assured, the feature vectors are partitioned according to the *rule* on x_k. For splitting, the three *rules* explained in Section 3.4.1 apply. Then, the algorithm is repeated recursively for

each child node on the portion of feature vectors that matches the test leading to the child.

3.4.3 Random Forest (RF-T)

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The formal definition states that a random forest is a classifier consisting of a collection of tree-structured classifiers $\{H(\mathbf{x}, \Theta_q),\ q = 1, 2, \ldots, Q\}$ where the $\{\Theta_q\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$ [77].

A random forest is constructed using the bagging method along with random feature selection. Given a training set X, the procedure starts with randomly forming bootstrap training sets $X_1,\ X_2, \ldots,\ X_Q$ and specifying a which is the parameter indicating the number of random features to select at a node. Although we use the same notation, this partitioning, in general, is different than the one in Section 3.4.1. Bagging corresponds to splitting the bootstrap training set into in-bag (two-thirds) and out-of-bag (one-third) portions. The *rule* at a node of the qth tree is defined by evaluating the normalized information gain explained in Section 3.4.1 using the a randomly selected features of the in-bag portion of bootstrap training set X_q and choosing the one with the highest gain. Then, the classifier $H(\mathbf{x}, \Theta_q)$, where $\Theta_q = (X_q, a)$, is constructed. Out-of-bag portion is used for estimating the generalization error which is the error rate of the classifier on the training set. In this regard, bagging resembles three-fold cross validation with the slight difference that three-fold cross validation is biased whereas out-of-bag estimates are unbiased. In the WEKA implementation, several other parameters such as strength, correlation, and variable importance that

are listed under out-of-bag estimates are missing. The only parameters specified are $a = 5$ and Q=10.

3.5 Gaussian Mixture Model (GMM)

In GMM, each feature vector in the training set is assumed to be associated with a mixture of M different and independent multi-variate Gaussian distributions. Expectation-Maximization (EM) algorithm is implemented to estimate the mean vector and the covariance matrices of the individual mixture components [78]. To define the iteration procedure, we start with a mixture model as a linear combination of M densities:

$$p(\mathbf{x}_i \mid \Upsilon) = \sum_{m=1}^{M} \alpha_m \, p_m(\mathbf{x}_i \mid \boldsymbol{\theta}_m) \tag{3.10}$$

where $\Upsilon = (\alpha_1, \ldots, \alpha_M; \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$ such that $\alpha_m \geq 0$ and $\sum_{m=1}^{M} \alpha_m = 1$. Analytical expressions for $\boldsymbol{\theta}_m$ can be obtained for the special case of GMM for which $\boldsymbol{\theta}_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$. Considering the GMM case, each distribution $p_m(\mathbf{x} \mid \boldsymbol{\theta}_m)$ is assumed to have a multi-variate Gaussian probability density function with mean $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$:

$$\begin{aligned} p_m(\mathbf{x} \mid \boldsymbol{\theta}_m) &= p_m(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \\ &= \frac{1}{(2\pi)^{N/2} \mid \boldsymbol{\Sigma}_m \mid^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m)\right] \end{aligned} \tag{3.11}$$

Starting with initial parameter estimates $\Upsilon^{(0)} = (\alpha_1^{(0)}, \ldots, \alpha_M^{(0)}; \boldsymbol{\theta}_1^{(0)}, \ldots, \boldsymbol{\theta}_M^{(0)})$, the elements of parameter vector Υ are updated recursively as follows:

$$\alpha_m{}^{(\kappa)} = \frac{1}{I} \sum_{i=1}^{I} p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)}) \tag{3.12}$$

$$\boldsymbol{\mu}_m{}^{(\kappa)} = \frac{\sum_{i=1}^{I} p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)}) \mathbf{x}_i}{\sum_{i=1}^{I} p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)})} \tag{3.13}$$

$$\boldsymbol{\Sigma}_m{}^{(\kappa)} = \frac{\sum_{i=1}^{I} p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)})(\mathbf{x}_i - \boldsymbol{\mu}_m{}^{(\kappa-1)})(\mathbf{x}_i - \boldsymbol{\mu}_m{}^{(\kappa-1)})^T}{\sum_{i=1}^{I} p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)})} \tag{3.14}$$

where

$$p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)}) = \frac{\alpha_m^{(\kappa-1)} p_m(\mathbf{x}_i \mid \boldsymbol{\theta}_m^{(\kappa-1)})}{\sum_{m=1}^{M} \alpha_m^{(\kappa-1)} p_m(\mathbf{x}_i \mid \boldsymbol{\theta}_m^{(\kappa-1)})} \tag{3.15}$$

Among the five types of covariance matrix provided in [78], the arbitrary one (Equation (3.14)) is used where each component in the mixture has a different covariance matrix with non-zero off-diagonal elements. The expressions provided here are valid for the generalized EM algorithm. Recursive iteration can be terminated if the change in the log-likelihood

$$\begin{aligned} \mathcal{E}(\Upsilon^{(\kappa)}, \Upsilon^{(\kappa-1)}) = & \sum_{m=1}^{M} \sum_{i=1}^{I} \log(\alpha_m)\, p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)}) \\ & + \sum_{m=1}^{M} \sum_{i=1}^{I} \log\big(p_m(\mathbf{x}_i \mid \boldsymbol{\theta}_m)\big)\, p(m \mid \mathbf{x}_i, \Upsilon^{(\kappa-1)}) \end{aligned} \tag{3.16}$$

for consecutive iterations is less than a preset threshold value or if the number of iterations exceeds the limit.

3.6 Support Vector Machines (SVM)

SVM technique is introduced by Vladimir Vapnik in the late seventies and it is being used intensively for complex classification tasks [79, 80, 81]. The general algorithm for SVM is explained below [82].

In SVM classification technique, it is desired to estimate a function $f : \mathbb{R} \rightarrow \{\pm 1\}$ using the training data. Given the training data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$ and the corresponding desired output labels $Z = \{z_1, z_2, \ldots, z_I\}$, we have a set of I training points:

$$O = \{(\mathbf{x}_i, z_i) \in \mathbb{R}^N \times \{-1, 1\}\} \qquad\qquad i = 1, 2, \ldots, I \tag{3.17}$$

where $\mathbf{x}_i$'s are the training feature vectors labeled with z_i as -1 or as $+1$ according to function $f(\mathbf{x}) = z$. Here, the problem is posed as a binary classification problem since WEKA builds a binary classifier in which, assuming there are c

classes in the actual training set, there exists $\frac{c(c-1)}{2}$ pairwise problems so that every pair of classes is considered [83]. Hyperplanes of the form

$$(\mathbf{v} \cdot \mathbf{x}) + b = 0 \qquad \mathbf{v} \in \mathbb{R}^N, b \in \mathbb{R} \tag{3.18}$$

are assigned to separate the pair of classes $\{-1, 1\}$. The form of the decision functions corresponding to these hyperplanes can be expressed as

$$f(\mathbf{x}) = \text{sign}\big[(\mathbf{v} \cdot \mathbf{x}_i) + b\big] \tag{3.19}$$

where $\mathbf{v}$ is the vector normal to the hyperplane and b is an arbitrary constant. It is desired to select $\mathbf{v}$ and b such that the margin between two parallel separating hyperplanes is maximum. These hyperplanes are given with the following equations:

$$(\mathbf{v} \cdot \mathbf{x}_i) + b \leq -1 \quad \text{for all } \mathbf{x}_i \text{ in class 1} \tag{3.20}$$

$$(\mathbf{v} \cdot \mathbf{x}_i) + b \geq 1 \quad \text{for all } \mathbf{x}_i \text{ in class 2} \tag{3.21}$$

These inequalities can be compactly combined into a single inequality:

$$z_i \cdot \big[(\mathbf{v} \cdot \mathbf{x}_i) + b\big] \geq -1 \tag{3.22}$$

A simple binary classification problem with a corresponding hyperplane solution is depicted in Figure 3.1. The margin that we want to maximize is measured to be $\frac{2}{\|\mathbf{v}\|}$ and $\|\mathbf{v}\|$ must be minimized to maximize that margin. To simplify the problem, the term, $\frac{1}{2}\|\mathbf{v}\|^2$ is minimized instead of $\|\mathbf{v}\|$. Using the inequality given in Equation (3.22) and the optimization constraint, we have the following quadratic programming (QP) optimization problem:

$$\text{minimize } \frac{1}{2}\|\mathbf{v}\|^2 \text{ subject to } z_i \cdot \big[(\mathbf{v} \cdot \mathbf{x}_i) + b\big] \geq 1 \qquad i = 1, 2, \ldots, I \tag{3.23}$$

A functional is constructed using the method of Lagrange multipliers to come up with a solution to the optimization problem presented.

$$\mathcal{L}(\mathbf{v}, b, \lambda) = \frac{1}{2}\|\mathbf{v}\|^2 - \sum_{i=1}^{I} \lambda_i \Big(z_i \cdot \big[(\mathbf{v} \cdot \mathbf{x}_i) + b\big] - 1 \Big) \tag{3.24}$$

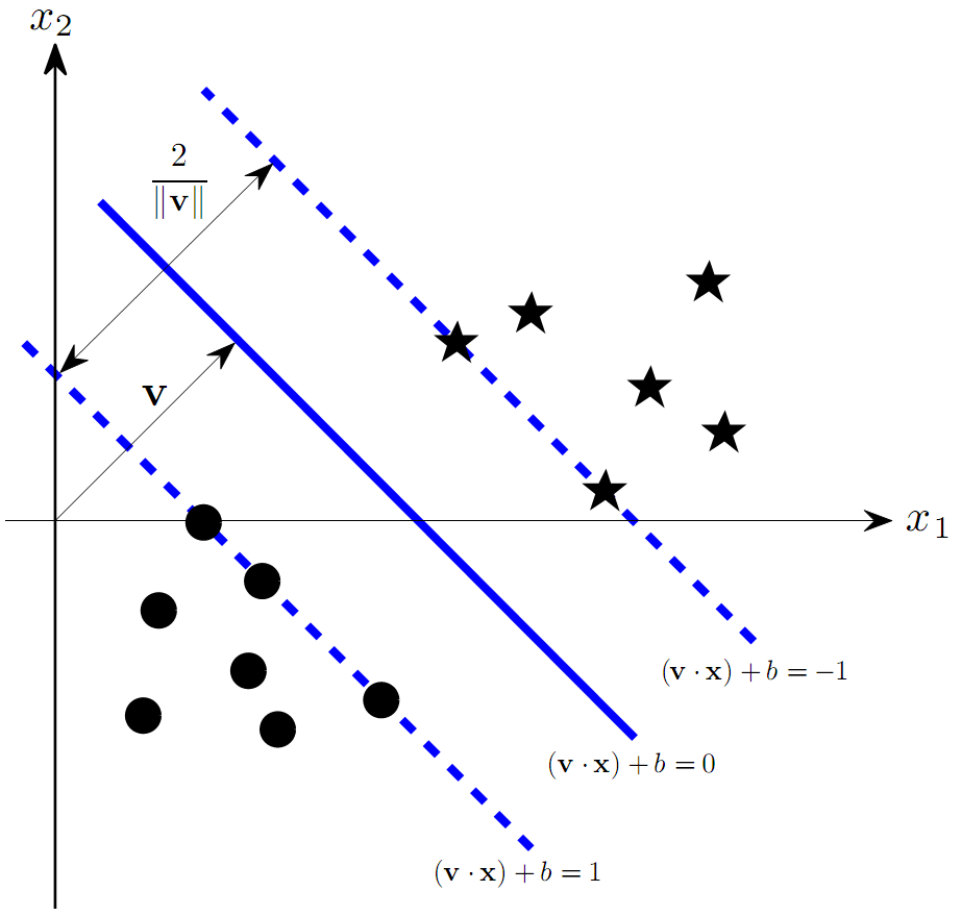


Figure 3.1: Simple binary classification problem. Three hyperplanes separate the balls from the stars. The hyperplane represented with a solid line is the separating hyperplane that is to be optimized. Two other hyperplanes represented with dashed lines and parallel to the separating hyperplane are the marginal hyperplanes.

The above Lagrangian must be minimized with respect to $\mathbf{v}$ and b and maximized with respect to $\lambda \geq 0$. To achieve that, we set the partial derivative of $\mathcal{L}(\mathbf{v},b,\lambda)$ with respect to $\mathbf{v}$ and b to zero and obtain $\sum_{i=1}^{I} \lambda_i z_i = 0$ and $\mathbf{v} = \sum_{i=1}^{I} \lambda_i z_i \mathbf{x}_i$. Solving these two equations simultaneously will yield several non-zero λ_i and using Karush-Kuhn-Tucker complementary condition:

$$\lambda_i \cdot \Big[z_i \cdot \big[(\mathbf{v} \cdot \mathbf{x}_i + b) \big] - 1 \Big] = 0 \qquad\qquad i = 1, 2, \ldots, I \qquad (3.25)$$

corresponding $\mathbf{x}_i$'s provided with non-zero λ_i will satisfy Equation (3.22) and be the support vectors through which the marginal hyperplanes shown in Figure 3.1 will pass. Substituting the expression of $\mathbf{v}$ into Equation (3.19), the decision

function giving the optimal hyperplane is

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{I} z_i \lambda_i (\mathbf{x} \cdot \mathbf{x}_i) + b\right) \tag{3.26}$$

The optimization problem stated in Equation (3.23) cannot be solved if the training data is not linearly separable. This issue is overcome by mapping the original training data to some other nonlinearly related dot product space using kernel functions. Once the mapping $\Psi : \mathbb{R}^N \rightarrow \mathcal{F}$ is performed, the algorithm provided is applied in $\mathcal{F}$ to find the optimal separating hyperplane. In this case, the expression given in Equation (3.26) can be rewritten as

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{I} z_i \lambda_i K(\mathbf{x},\mathbf{x}_i) + b\right) \tag{3.27}$$

where $K(\mathbf{x},\mathbf{x}_i) = \Psi(\mathbf{x}) \cdot \Psi(\mathbf{x}_i)$. In our experiments, Gaussian radial basis function of the form $K(\mathbf{x},\mathbf{x}_i) = e^{-\gamma||\mathbf{x}-\mathbf{x}_i||^2}$ is employed as the Kernel. In order to decide which Kernel to use, we tested SVM classifier with various Kernels and different parameters. The Kernels that are implemented are: polynomial Kernel function $K(\mathbf{x},\mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i)^\eta$ for $\eta \in \{1, 2, 3, 4\}$, normalized polynomial Kernel function $K(\mathbf{x},\mathbf{x}_i) = \frac{(\mathbf{x} \cdot \mathbf{x}_i)}{\sqrt{||\mathbf{x}||^2 + ||\mathbf{x}_i||^2}}$, and Gaussian radial basis function $K(\mathbf{x},\mathbf{x}_i) = e^{-\gamma||\mathbf{x}-\mathbf{x}_i||^2}$ for $\gamma \in \{2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^0, 2^1, 2^3, 2^5\}$ and $C \in \{2^{-5}, 2^{-3}, 2^{-1}, 2^0, 2^1, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}\}$ where C is the soft margin parameter also called the complexity parameter. Every combination of γ and C is considered. The SVM classifier is tested based on the 5-fold cross validation using the one third of the original dataset and L1O cross validation using the whole dataset. The radial basis function with $\gamma = 2$ and $C = 2$ has provided the best classification performance and used in the actual tests.

SVM implemented in WEKA is enhanced with sequential minimal optimization (SMO) algorithm. SMO breaks down the QP problem mentioned earlier (Equation (3.23)) to smallest possible QP problems that can be solved analytically. Resulting SVM is improved in terms of computational cost and scaling [84].

Chapter 4

Experimental Results

In this chapter, experimental results are presented and compared considering the cross-validation techniques, the correct differentiation rates, the confusion matrices, the machine learning environments, the previous results, and the computational considerations. The main purpose of this chapter is to determine the best classifier to be used in activity classification. It is also intended to determine the most informative sensor type and sensor unit location on the body. In order to achieve that, the experimental results systematically consider all possible combinations of sensor types and sensor unit locations on the body. In addition, the activities that are confused with each other are indicated, the comparison between the machine learning environments, WEKA and PRTools, is given, the results of the previous study [57] are recalled and compared with the results of this study, and finally, computational requirements of the classification techniques are considered.

4.1 Cross-Validation Techniques

The classification techniques described in Chapter 3 are employed to classify the 19 different activities using the 30 features selected by PCA. A total of $9,120$ ($=$ 60 feature vectors $\times$ 19 activities $\times$ 8 subjects) feature vectors are available, each containing the 30 reduced features of the 5-s signal segments. In the training and testing phases of the classification methods, we use the repeated random sub-sampling (RRSS), P-fold, and leave-one-out (L1O) cross-validation techniques.

In RRSS, we divide the 480 feature vectors from each activity type randomly into two sets so that the first set contains 320 feature vectors (40 from each subject) and the second set contains 160 (20 from each subject). Therefore, two thirds (6,080) of the 9,120 feature vectors are used for training and one third (3,040) for testing. This is repeated 10 times and the resulting correct differentiation percentages are averaged. The disadvantage of this method is that some observations may never be selected in the testing or the validation phase, whereas others may be selected more than once. In other words, validation subsets may overlap.

In P-fold cross validation, the 9,120 feature vectors are divided into $P = 10$ partitions, where the 912 feature vectors in each partition are selected completely randomly, regardless of the subject or the class they belong to. One of the P partitions is retained as the validation set for testing, and the remaining $P - 1$ partitions are used for training. The cross-validation process is then repeated P times (the folds), where each of the P partitions is used exactly once for validation. The P results from the folds are then averaged to produce a single estimation. The random partitioning is repeated 10 times and the average correct differentiation percentage is reported. The advantage of this validation method over RRSS is that all feature vectors are used for both training and testing, and each feature vector is used for testing exactly once in each of the 10 runs.

Finally, we also used subject-based L1O cross validation, where the $7,980$ ($=$ 60 vectors $\times$ 19 activities $\times$ 7 subjects) feature vectors of seven of the subjects are used for training and the 1,140 feature vectors of the remaining subject are used in turn for validation. This is repeated eight times such that the feature vector set of each subject is used once as the validation data. The eight correct classification rates are averaged to produce a single estimate. This is similar to P-fold cross validation with P being equal to the number of subjects ($P = 8$), and where all the feature vectors in the same partition are associated with the same subject.

4.2 Correct Differentiation Rates

The algorithms for the techniques used in this study are provided on two commonly used open source environments: WEKA, a Java-based software [60]; and PRTools, a MATLAB toolbox [61]. The NB and ANN classifiers are tested in both of these software environments to compare two different implementations of the algorithms and the environments themselves. SVM and decision-tree techniques, namely, NB-T, J48-T, and RF-T are tested using WEKA. PRTools is used for testing DBC and GMM for different cases where the number of mixtures in the model varies from one to four.

The classification techniques are tested based on every combination of sensor types (gyro, acc, and mag) and different sensor units (T, RA, LA, RL, LL). In the first approach, training data extracted from all possible combinations of sensor types are used for classification and correct differentiation rates and standard deviations over 10 runs are provided in Tables 4.1–4.3. Because L1O cross validation would give the same classification percentage if the complete cycle over the subject-based partitions is repeated, its standard deviation is zero. Correct differentiation rates are also depicted in the form of bar graphs in Figure 4.1 for

better visualization. In the second approach, training data extracted from all possible combinations of different sensor units are used for the tests and correct differentiation rates are tabulated in Tables 4.4 and 4.5. Each cross-validation technique is applied in these tests.

It is observed that the 10-fold cross validation has the best performance, with RRSS following it with slightly smaller rates. The difference is caused by the fact that in 10-fold cross validation, a larger data set is used for training. On the other hand, L1O has the smallest rates in all cases because each subject performs the activities in a different manner. Outcomes obtained by implementing L1O indicate that the dataset should be sufficiently comprehensive in terms of the diversity of the physical characteristics of the subjects. Each additional subject with distinctive characteristics included in the initial feature vector set will improve the correct classification rate of novel feature vectors.

Compared to other decision-tree methods, the random forest outperforms in all of the cases. Such an outcome is expected since the random forest consists of 10 decision trees each voting individually for a certain class and the class with the highest vote is classified to be the correct one. Despite its random nature, it competes with the other classifiers and achieves the average correct differentiation rate of 98.6% for 10-fold cross validation when data from all sensors is used (Table 4.3). NB-T method seems to be the worst of all decision trees because of its independence assumption. WEKA provides a large number of decision-tree methods to choose from. However, some of these such as the best-first and logistic model decision-tree classifiers are not applicable in our case because of the size of the training set, especially for 10-fold cross validation.

Generally, the best performance is expected from ANN and SVM for problems involving multi-dimensional and continuous feature vectors [85]. L1O cross-validation results for each sensor combination indicate that they have a great

capacity for generalization. As a consequence, they are less susceptible to overfitting than every other classifier, especially, the GMM. ANN and SVM classifiers are the best classifiers among all and usually have slightly higher performance than GMM_1 (99.1%) with 99.2% for 10-fold cross validation when the feature vectors extracted from combination of all sensors are used for classification (Table 4.3). In the case of L1O cross validation, their success rates are significantly better than GMM_1.

The ANN classifier implemented in PRTools seems to be quite incompetent. In an ANN trained with the back-propagation algorithm, the system should be initialized with proper parameters. The most important parameters are the learning and momentum constants and initial values of the connection weights. PRTools does not allow user to set the values for the learning and momentum constants which play a crucial role in updating the weights. Without proper values set for these constants, it is difficult to provide the system with suitable initial weights. Therefore, the correct differentiation rates regarding ANN implemented in PRTools do not reflect the true potential of the classifier.

Considering the outcomes obtained based on 10-fold cross validation and each sensor combination, it is difficult to determine the number of mixture components to be used in the GMM method. The average correct differentiation rates are quite close to each other for GMM_1, GMM_2, and GMM_3 (Gaussian mixture models with one, two, and three components). However, in case of RRSS and especially L1O cross validation, the rates rapidly decrease as the number of components in the mixture increases. Such an outcome is not anticipated. It seems that the data set is not sufficiently large to train GMM with a mixture of multiple components. Indeed, it is observed in Table 4.1(c) that the GMM_3 and GMM_4 could not be trained due to insufficient data. Another interpretation of the results would be overfitting [67]. While multiple Gaussian estimators are exceptionally complex for classification of training patterns, they are unlikely

to result in acceptable classification of novel patterns. Low differentiation rates of GMM for L1O in all cases support the overfitting condition. Despite the incompetent outcomes taken from this method for L1O case, it is the third best classifier with 99.1% average correct differentiation rate based on 10-fold cross validation when data from all sensors is used (Table 4.3).

The comparison of classification results based on each sensor combination reveal quite an unexpected outcome. It seems that when the data set corresponding to magnetometer alone is used, the average correct differentiation rate is higher than the rates provided by the other two sensor types used alone. For a considerable number of classification methods, the rate provided by magnetometer data alone outperforms the rates provided by the other two sensors combined together. It can be observed in Figure 4.1 that for almost all classification methods applied based on all cross-validation techniques, the turquoise bar is higher than the green bar at the top plots of the figures except for the GMM model used in L1O cross validation. It can be stated that the features extracted from magnetometer data, which is slowly varying in nature, are not sufficiently diverse for the training of the GMM classifier. This statement is supported by the results provided in Table 4.1(c) such that GMM_3 and GMM_4 cannot be trained with magnetometer-based feature vectors. The best performance (98.8%) based on magnetometer data is achieved with SVM using 10-fold cross validation (Table 4.1(c)).

Correct differentiation rates obtained by using feature vectors based on gyroscope data are the worst. Outcomes of the combination of gyroscope with other two sensors are also usually worse than the combination of accelerometer and magnetometer. The magnetometers used in this study measure the strength of the magnetic field along three orthogonal axes and the combination of the quantities measured with respect to each axis provides the direction of the Earth's magnetic north. In other words, the magnetometers function as a compass. Thus,

the results discussed here indicate that the most useful source of information is provided by the feature vectors based on the compass data (magnetometer), then translational data (accelerometer), and finally, the rotational data (gyroscope). In general, the combination of these three types of data provides the best classification performance.

The case in which classifiers are tested based on combination of different sensor units (Table 4.4 and 4.5) shows that GMM usually has the best classification performance for all cross-validation techniques other than L1O. In L1O cross validation, ANN and SVM classifiers have the best performances. In 10-fold cross validation (Table 4.5(b)), correct differentiation rates achieved with GMM_2 are better than GMM_1 in tests for which single unit or combination of two units is used. Another remark regarding the contribution of each sensor unit is that the units placed on the legs (RL and LL) seem to provide the most useful data. Comparing the cases where feature vectors extracted from single sensor unit data, it is observed that highest correct classification rates are achieved with these two units. They improve the performance of the combinations in which they are used as well.

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		66.7±0.45	67.4±0.15	56.9
	ANN		79.8±0.71	84.3±0.17	60.9
	SVM		80.1±0.43	84.7±0.14	61.2
	tree methods	NB-T	62.3±1.22	67.8±0.73	36.4
		J48-T	61.9±0.66	68.0±0.35	45.2
		RF-T	73.1±0.58	78.3±0.34	53.3
PRTools	NB		63.9±0.67	67.7±0.30	49.7
	ANN		59.9±5.38	59.5±0.89	48.6
	DBC		68.5±0.81	69.7±0.30	56.9
	GMM	GMM_1	79.8±0.50	82.2±0.14	57.1
		GMM_2	76.8±0.82	83.4±0.26	42.5
		GMM_3	71.4±1.30	83.1±0.24	37.3
		GMM_4	64.7±1.39	82.6±0.25	32.1

(a)

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		80.5±0.67	80.8±0.09	73.6
	ANN		92.5±0.51	95.3±0.07	79.7
	SVM		91.2±0.61	94.6±0.09	81.0
	tree methods	NB-T	74.8±1.42	79.0±0.61	55.9
		J48-T	75.8±0.85	80.9±0.33	62.8
		RF-T	86.0±0.51	89.7±0.16	72.2
PRTools	NB		77.3±0.66	81.2±0.22	66.5
	ANN		76.2±2.58	75.4±1.29	67.5
	DBC		81.9±0.52	82.2±0.26	74.6
	GMM	GMM_1	93.3±0.48	95.1±0.07	74.8
		GMM_2	90.7±0.66	95.5±0.12	58.2
		GMM_3	86.0±1.31	95.3±0.13	53.0
		GMM_4	77.4±1.37	94.8±0.25	44.2

(b)

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		89.0±0.37	89.5±0.08	79.3
	ANN		97.5±0.28	98.6±0.06	81.5
	SVM		98.1±0.09	98.8±0.04	84.8
	tree methods	NB-T	90.9±0.85	94.3±0.33	52.3
		J48-T	90.0±0.60	93.8±0.15	65.8
		RF-T	96.9±0.25	98.1±0.12	78.2
PRTools	NB		91.9±0.36	93.5±0.17	74.1
	ANN		90.2±2.07	89.6±0.97	78.3
	DBC		91.0±0.88	92.0±0.33	82.6
	GMM	GMM_1	96.2±0.33	96.5±0.04	42.6
		GMM_2	96.2±0.47	97.3±0.18	22.6
		GMM_3	94.2±0.87	–	–
		GMM_4	89.8±1.54	–	–

(c)

Table 4.1: Correct differentiation rates and the standard deviations based on all classification techniques, cross-validation methods, and both environments. Only (a) gyroscopes, (b) accelerometers, (c) magnetometers are used for classification.

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		84.6±0.55	85.2±0.13	76.3
	ANN		95.1±0.24	96.9±0.09	82.6
	SVM		95.0±0.19	96.7±0.07	83.3
	tree methods	NB-T	81.7±1.66	86.5±0.37	57.2
		J48-T	82.5±1.11	87.0±0.19	66.2
		RF-T	91.9±0.43	94.5±0.15	77.7
PRTools	NB		84.2±0.36	86.9±0.16	71.2
	ANN		81.2±3.85	81.4±1.36	71.9
	DBC		85.3±0.71	86.2±0.41	76.5
	GMM	GMM_1	96.0±0.30	97.1±0.10	79.3
		GMM_2	93.9±0.42	96.9±0.07	50.7
		GMM_3	88.9±0.90	96.7±0.11	44.1
		GMM_4	81.5±1.70	96.4±0.15	37.9

(a)

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		92.1±0.27	92.2±0.09	85.1
	ANN		98.5±0.14	99.0±0.04	87.5
	SVM		98.6±0.16	99.0±0.04	86.1
	tree methods	NB-T	92.0±0.69	94.9±0.30	61.3
		J48-T	90.7±1.27	94.5±0.15	75.0
		RF-T	97.5±0.22	98.4±0.06	81.8
PRTools	NB		93.8±0.43	95.4±0.09	77.2
	ANN		91.6±2.59	91.4±1.28	84.6
	DBC		92.9±0.64	93.0±0.47	85.2
	GMM	GMM_1	98.6±0.12	98.9±0.03	64.6
		GMM_2	96.9±1.68	98.7±0.06	35.9
		GMM_3	93.3±1.04	98.7±0.06	29.8
		GMM_4	86.1±3.43	98.6±0.09	26.1

(b)

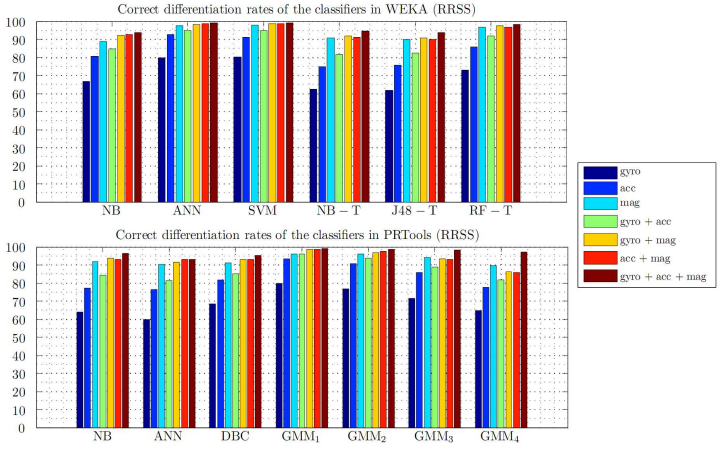
	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		92.8±0.41	92.7±0.09	87.2
	ANN		98.7±0.15	99.2±0.04	92.2
	SVM		98.5±0.12	99.0±0.03	89.5
	tree methods	NB-T	91.1±0.81	93.7±0.31	64.9
		J48-T	89.9±0.55	93.1±0.12	79.8
		RF-T	96.9±0.26	98.1±0.10	86.0
PRTools	NB		93.1±0.50	94.1±0.07	81.8
	ANN		93.0±1.97	92.1±0.82	87.1
	DBC		93.2±0.70	93.5±0.24	85.7
	GMM	GMM_1	98.7±0.23	99.1±0.03	69.8
		GMM_2	97.5±1.01	99.0±0.06	46.9
		GMM_3	93.2±1.85	98.8±0.07	39.8
		GMM_4	85.9±4.67	98.6±0.09	34.0

(c)

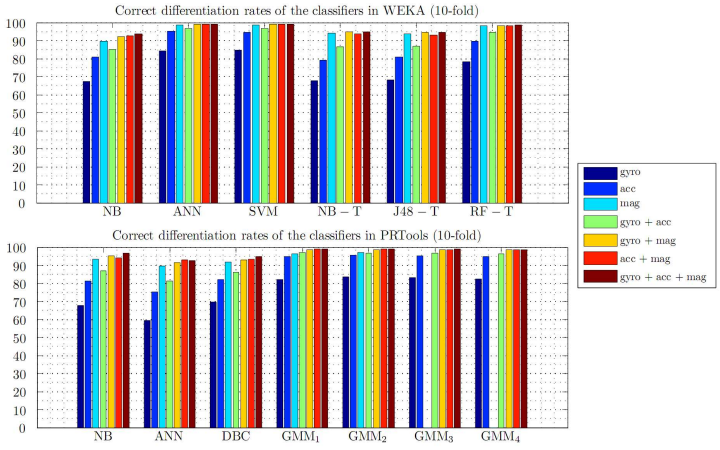
Table 4.2: Correct differentiation rates and the standard deviations based on all classification techniques, cross-validation methods, and both environments. Two types of sensors, namely, (a) gyroscopes and accelerometers, (b) gyroscopes and magnetometers, (c) accelerometers and magnetometers are used for classification.

	classification techniques		cross validation		
			RRSS	**10-fold**	**L1O**
WEKA	NB		93.9±0.49	93.7±0.08	89.2
	ANN		99.1±0.13	99.2±0.05	91.0
	SVM		99.1±0.09	99.2±0.03	89.9
	tree methods	NB-T	94.6±0.68	94.9±0.16	67.7
		J48-T	93.8±0.73	94.5±0.17	77.0
		RF-T	98.3±0.24	98.6±0.05	86.8
PRTools	NB		96.5±0.46	96.6±0.07	83.8
	ANN		93.0±3.05	92.5±1.61	84.2
	DBC		94.7±0.60	94.8±0.16	89.0
	GMM	GMM_1	99.1±0.20	99.1±0.02	76.4
		GMM_2	98.8±0.17	99.0±0.03	48.1
		GMM_3	98.2±0.30	98.9±0.07	37.6
		GMM_4	97.3±0.37	98.8±0.07	37.0

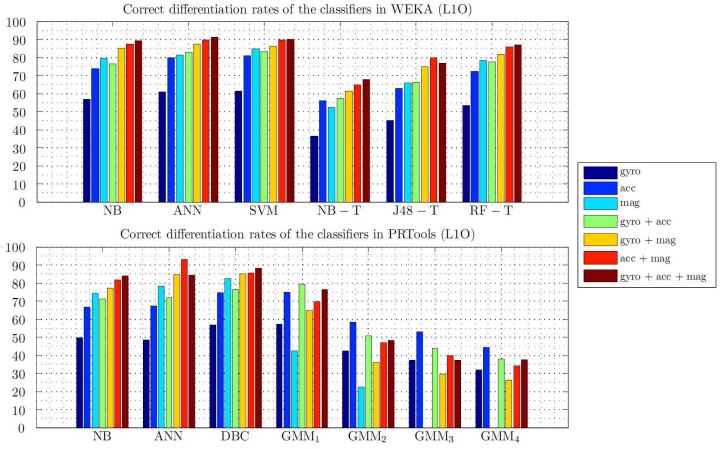
Table 4.3: Correct differentiation rates and the standard deviations based on all classification techniques, cross-validation methods, and both environments. All sensors are used for classification.



(a)



(b)



(c)

Figure 4.1: Comparison of classifiers and combinations of different sensor types in terms of correct differentiation rates using (a) RRSS, (b) 10-fold, (c) L1O cross validation. The patterns in the legends are ordered from left to right in the bar chart.

units used	NB	ANN	SVM	NB-T	J48-T	RF-T		NB	ANN	SVM	NB-T	J48-T	RF-T
–	–	–	–	–	–	–	+T	70.5	92.2	91.0	69.8	70.5	83.5
RA	67.5	92.1	91.2	73.3	71.9	84.4	+T	81.7	95.8	96.1	81.9	79.4	92.1
LA	69.2	89.7	92.7	68.2	69.3	83.1	+T	82.4	96.0	96.3	81.9	79.2	92.8
RL	87.0	95.7	94.9	80.5	82.1	89.9	+T	86.2	97.2	97.0	85.1	85.3	94.3
LL	86.3	96.9	95.6	82.4	84.0	90.0	+T	85.8	97.5	97.0	84.2	84.8	94.1
RA+LA	78.4	94.0	95.6	81.7	79.7	91.8	+T	87.0	97.3	97.4	87.8	85.4	95.2
RL+LL	88.8	97.3	96.8	87.6	88.8	94.9	+T	90.8	97.9	97.9	90.5	89.7	96.4
RA+RL	89.0	97.1	97.2	86.5	86.2	94.6	+T	90.1	97.9	98.2	88.6	87.4	96.3
LA+LL	89.7	97.6	97.5	85.7	86.6	94.7	+T	91.8	98.1	98.0	87.9	87.8	96.2
RA+LL	88.9	97.7	97.6	85.9	85.7	94.7	+T	90.3	98.0	98.1	88.4	87.8	96.2
LA+RL	90.3	97.4	97.5	85.8	85.3	94.2	+T	90.3	97.9	98.0	88.1	86.6	96.3
RA+LA+RL	90.9	98.3	98.3	88.4	87.6	96.3	+T	92.9	98.4	98.7	90.5	88.8	97.1
RA+LA+LL	90.8	98.0	98.4	89.6	88.8	96.6	+T	92.3	98.4	98.6	90.2	89.5	97.0
RA+RL+LL	91.1	98.2	98.1	90.1	89.6	96.7	+T	92.5	98.5	98.7	91.2	90.8	97.3
LA+RL+LL	91.6	98.2	98.2	91.3	90.7	97.1	+T	92.7	98.2	98.5	91.3	91.3	97.6
RA+LA+RL+LL	92.2	98.7	98.7	91.5	90.6	97.7	+T	93.9	99.1	99.1	94.6	93.8	98.3

(a)

units used	NB	ANN	SVM	NB-T	J48-T	RF-T		NB	ANN	SVM	NB-T	J48-T	RF-T
–	–	–	–	–	–	–	+T	71.5	95.3	95.7	77.5	78.2	89.4
RA	67.3	95.5	95.1	80.5	79.7	89.8	+T	82.7	97.5	97.8	87.1	86.1	95.2
LA	70.0	92.6	96.2	76.0	76.6	88.5	+T	83.5	97.7	97.9	87.8	86.3	95.7
RL	87.5	97.6	96.8	86.1	86.3	93.2	+T	86.4	98.4	98.3	90.0	89.4	96.4
LL	87.0	98.2	97.6	87.1	87.7	93.2	+T	86.1	98.6	98.4	89.9	89.9	96.6
RA+LA	79.1	95.5	97.5	87.4	86.3	94.9	+T	87.9	98.0	98.5	91.4	90.0	97.0
RL+LL	89.0	98.5	98.1	91.2	92.1	96.6	+T	91.0	98.8	98.8	93.3	93.1	97.7
RA+RL	89.2	97.8	98.4	90.4	90.6	96.7	+T	90.5	98.5	98.8	92.7	91.3	97.6
LA+LL	90.2	98.5	98.5	90.1	90.1	96.5	+T	92.2	98.6	98.8	92.0	92.0	97.7
RA+LL	89.2	98.6	98.5	90.3	90.1	96.7	+T	90.8	98.8	98.9	92.7	91.8	97.6
LA+RL	90.6	98.4	98.4	90.3	89.5	96.4	+T	91.2	98.7	98.7	92.5	91.4	97.7
RA+LA+RL	91.1	98.9	99.0	92.2	92.0	97.8	+T	93.2	98.9	99.1	94.1	92.8	98.1
RA+LA+LL	90.9	98.7	99.0	92.8	92.3	97.8	+T	92.4	99.0	99.1	93.2	93.1	98.1
RA+RL+LL	91.2	98.9	98.9	93.7	93.0	97.9	+T	93.0	98.9	99.1	94.6	94.3	98.4
LA+RL+LL	91.5	98.7	98.9	94.0	93.7	98.0	+T	93.0	98.9	99.0	94.6	94.3	98.4
RA+LA+RL+LL	92.4	99.1	99.1	95.0	94.3	98.6	+T	93.7	99.2	99.2	94.9	94.5	98.6

(b)

units used	NB	ANN	SVM	NB-T	J48-T	RF-T		NB	ANN	SVM	NB-T	J48-T	RF-T
–	–	–	–	–	–	–	+T	58.8	67.1	70.3	40.2	49.1	58.7
RA	57.8	64.2	67.6	36.0	43.6	55.9	+T	71.9	78.4	80.5	46.2	57.8	69.7
LA	55.6	64.3	65.5	37.9	42.9	55.8	+T	73.9	77.6	80.4	46.2	56.1	69.6
RL	78.5	81.7	83.4	65.0	67.7	77.2	+T	78.6	83.5	85.6	58.0	66.6	78.9
LL	78.6	82.7	84.1	60.3	70.5	76.8	+T	78.5	86.1	87.6	61.1	66.9	80.1
RA+LA	66.7	75.5	76.3	42.7	52.6	65.7	+T	76.5	82.9	83.9	48.7	65.8	76.4
RL+LL	81.9	85.6	86.2	65.0	76.4	83.2	+T	83.6	89.4	89.0	68.3	75.7	84.5
RA+RL	83.0	84.3	86.3	66.4	72.4	81.5	+T	84.7	88.4	88.5	60.8	72.1	83.0
LA+LL	83.3	83.6	84.8	61.2	70.9	81.3	+T	84.7	86.5	87.0	60.4	71.6	83.3
RA+LL	82.5	86.1	85.4	58.6	70.0	79.7	+T	83.4	89.5	88.9	61.4	71.6	83.0
LA+RL	83.2	85.7	84.9	59.9	72.1	80.3	+T	83.7	87.5	86.6	59.1	73.5	82.2
RA+LA+RL	84.7	85.5	86.0	61.7	72.6	82.5	+T	86.2	88.9	88.4	60.5	74.8	85.6
RA+LA+LL	84.5	85.6	85.6	65.4	73.0	81.1	+T	86.7	89.5	89.1	63.7	72.8	86.1
RA+RL+LL	85.6	86.6	86.7	66.5	76.3	84.3	+T	86.7	90.6	89.8	65.9	76.4	86.5
LA+RL+LL	84.8	86.7	85.8	68.2	77.4	85.2	+T	86.8	88.5	88.7	66.3	78.4	87.0
RA+LA+RL+LL	86.8	86.1	86.4	67.3	78.4	86.2	+T	89.2	91.0	89.9	67.7	77.0	86.8

(c)

Table 4.4: All possible sensor unit combinations and the corresponding correct classification rates for classification methods in WEKA using (a) RRSS, (b) 10-fold, (c) L1O cross validation.

units used	NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4		NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4
–	–	–	–	–	–	–	–	+T	67.9	71.4	77.8	93.9	91.1	85.8	73.1
RA	68.3	66.1	76.1	91.9	88.1	84.3	76.7	+T	82.2	77.4	85.9	96.7	94.4	88.7	80.8
LA	67.7	68.6	76.2	92.2	90.1	84.5	76.2	+T	83.9	83.0	86.5	96.7	93.9	85.7	75.4
RL	83.9	85.2	86.3	95.8	94.6	91.8	85.4	+T	85.7	84.7	88.7	98.0	95.8	92.6	85.1
LL	83.9	82.4	85.3	96.6	95.8	92.7	84.7	+T	85.9	87.4	88.4	98.2	95.5	91.7	85.2
RA+LA	79.0	76.3	83.7	95.7	93.0	87.5	80.3	+T	89.4	85.5	88.3	97.6	94.9	89.6	82.0
RL+LL	88.3	86.4	89.3	97.9	96.8	93.6	85.4	+T	91.3	90.8	91.5	98.7	97.5	94.1	86.8
RA+RL	88.0	84.4	89.1	97.9	96.6	92.5	84.8	+T	92.0	88.9	91.4	98.4	96.5	92.6	80.1
LA+LL	88.7	86.3	89.4	97.9	96.5	92.1	85.6	+T	92.1	90.7	91.9	98.3	96.8	91.5	84.0
RA+LL	87.8	88.5	89.3	97.9	96.7	92.9	85.5	+T	92.5	89.7	91.8	98.3	97.0	92.8	84.9
LA+RL	88.3	90.3	89.8	97.6	96.1	90.9	85.1	+T	90.9	89.8	91.5	98.2	96.3	92.1	86.1
RA+LA+RL	90.8	87.7	91.4	98.3	96.7	91.9	83.3	+T	93.8	91.4	92.9	98.6	96.8	91.5	84.5
RA+LA+LL	91.0	89.0	91.4	98.3	97.3	91.4	85.1	+T	93.4	91.4	93.3	98.7	97.2	92.0	86.3
RA+RL+LL	91.5	91.0	91.4	98.6	97.3	94.4	87.0	+T	93.7	93.1	93.6	98.7	97.8	93.0	86.1
LA+RL+LL	92.7	92.0	91.7	98.6	97.6	94.2	87.8	+T	94.4	93.0	93.8	98.7	97.6	93.0	86.9
RA+LA+RL+LL	93.1	92.4	93.0	98.9	97.3	93.9	87.0	+T	96.5	93.0	94.7	99.1	98.8	98.2	97.3

(a)

units used	NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4		NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4
–	–	–	–	–	–	–	–	+T	73.5	71.2	79.5	95.2	96.3	95.9	95.4
RA	72.8	67.7	77.3	93.3	94.8	94.8	94.2	+T	86.2	80.0	86.8	97.7	97.3	97.1	96.7
LA	72.5	68.0	77.1	93.5	95.1	95.0	94.6	+T	87.8	81.5	87.3	97.5	97.5	97.3	96.8
RL	87.0	84.5	87.1	96.7	97.4	97.3	97.0	+T	88.1	85.0	89.1	98.4	98.5	98.3	98.1
LL	87.7	82.6	86.3	97.4	97.7	97.6	97.3	+T	89.3	86.1	89.3	98.7	98.7	98.4	98.3
RA+LA	83.9	76.2	84.5	96.8	96.9	96.8	96.1	+T	91.6	84.3	89.6	98.3	97.9	97.7	97.4
RL+LL	91.3	87.7	89.6	98.5	98.5	98.3	98.0	+T	93.3	89.8	92.0	99.0	98.8	98.7	98.6
RA+RL	90.3	85.9	89.6	98.3	98.4	98.2	97.9	+T	93.8	88.7	92.0	98.6	98.6	98.4	98.2
LA+LL	91.3	88.6	90.1	98.4	98.5	98.3	98.1	+T	94.0	90.5	92.4	98.8	98.6	98.6	98.4
RA+LL	90.4	87.0	89.9	98.5	98.4	98.2	97.9	+T	94.1	89.6	92.1	98.8	98.6	98.5	98.4
LA+RL	91.1	88.1	90.5	98.2	98.4	98.2	98.0	+T	93.4	90.5	92.4	98.6	98.5	98.4	98.2
RA+LA+RL	92.7	88.0	91.8	98.8	98.7	98.5	98.2	+T	95.1	90.2	93.4	98.9	98.7	98.6	98.4
RA+LA+LL	93.0	88.8	91.8	98.8	98.7	98.6	98.4	+T	94.8	91.2	93.3	99.0	98.8	98.6	98.3
RA+RL+LL	93.3	89.4	92.2	98.9	98.8	98.7	98.5	+T	94.6	91.8	94.0	99.0	98.8	98.8	98.6
LA+RL+LL	94.3	88.8	91.9	98.9	98.8	98.8	98.6	+T	95.7	92.4	94.0	99.0	98.9	98.9	98.7
RA+LA+RL+LL	94.4	91.5	93.2	99.1	99.0	99.0	98.8	+T	96.6	92.5	94.8	99.1	99.0	98.9	98.8

(b)

units used	NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4		NB	ANN	DBC	GMM_1	GMM_2	GMM_3	GMM_4
–	–	–	–	–	–	–	–	+T	53.1	60.2	66.0	48.9	30.4	25.7	23.4
RA	48.7	59.8	63.4	44.2	26.2	20.8	23.5	+T	64.8	70.5	74.5	60.7	30.5	23.6	21.5
LA	50.3	57.2	59.8	45.7	33.2	27.0	21.0	+T	61.8	73.9	75.8	63.3	42.2	30.8	25.3
RL	75.6	78.4	79.7	71.1	55.5	50.0	47.2	+T	73.8	79.7	81.3	71.2	51.7	41.1	37.6
LL	72.7	75.2	78.4	70.1	57.4	53.6	48.8	+T	74.4	76.7	81.6	70.9	46.4	42.2	29.8
RA+LA	58.6	66.4	71.4	54.0	31.2	22.8	18.5	+T	66.8	74.7	79.2	65.0	37.2	31.6	22.4
RL+LL	79.3	80.7	83.8	73.9	57.6	52.5	47.8	+T	79.8	83.7	86.0	73.6	47.3	43.2	39.9
RA+RL	78.4	81.4	81.7	73.7	49.4	42.1	34.9	+T	77.6	82.0	85.4	75.0	46.5	39.0	33.4
LA+LL	78.5	79.9	82.2	72.2	50.9	39.0	31.0	+T	76.8	83.5	84.9	71.4	46.6	40.8	29.1
RA+LL	76.1	79.7	83.4	72.2	42.1	35.4	29.5	+T	77.9	82.7	84.0	75.4	43.2	35.8	28.9
LA+RL	77.7	81.5	82.0	73.2	54.6	44.8	36.7	+T	78.0	83.1	84.5	73.2	46.9	43.1	39.9
RA+LA+RL	75.9	81.4	85.2	73.3	46.5	42.5	29.8	+T	79.7	83.4	85.7	72.2	43.5	41.1	34.0
RA+LA+LL	77.5	81.3	84.4	74.2	45.0	37.0	26.8	+T	79.2	82.8	87.5	75.4	46.6	33.6	24.9
RA+RL+LL	80.0	82.9	85.3	73.9	51.3	43.8	35.9	+T	82.4	85.3	87.4	75.1	48.7	39.4	36.4
LA+RL+LL	79.3	83.8	86.4	74.1	51.4	44.2	36.0	+T	82.6	87.1	87.8	74.4	47.5	45.0	37.7
RA+LA+RL+LL	80.7	85.4	86.5	74.3	46.0	43.2	36.1	+T	83.8	84.2	89.0	76.4	48.1	37.6	37.0

(c)

Table 4.5: All possible sensor unit combinations and the corresponding correct classification rates for classification methods in PRTools using (a) RRSS, (b) 10-fold, (c) L1O cross validation.

4.3 Confusion Matrices

In order to show which activities are confused with each other, the confusion matrices of the different techniques are presented in Tables 4.6 and 4.7. We chose to employ the 10-fold cross-validation technique to report these results. Inspecting the confusion matrices of the different techniques, it can be observed that A7 and A8 are the activities most confused with each other. This is because both of these activities are performed in the elevator and the signals recorded from these activities have similar segments. Therefore, confusion at the classification stage becomes inevitable. A2 and A7, A13 and A14, as well as A9, A10, and A11 are also confused from time to time for similar reasons. The two activities that are almost never confused are A12 and A17.

The results for the confusion matrices given in Tables 4.6 and 4.7 are summarized in Table 4.9 to report the performance of the classifiers on distinguishing each activity. The feature vectors that belong to A3, A4, A5, A6, A12, A15, A17, and A18 are classified with the above average performance by all classifiers. The rest of the feature vectors cannot be classified well by some classifiers.

For ANN implemented in PRTools, since the network classifies some samples as belonging to none of the classes and output neurons take continuous values between 0 and 1, it is not possible to form a confusion matrix. The number of correctly and incorrectly classified feature vectors with 10-fold cross validation is given in Table 4.8. On the other hand, ANN implemented in WEKA assigns each feature vector to a certain class therefore it is possible to form a confusion matrix for that method (Table 4.6(b)).

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	445	0	32	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0
A2	3	437	0	0	0	0	37	3	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
A4	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	477	0	0	0	2	0	1	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	463	0	17	0	0	0	0	0	0	0	0	0	0	0
A7	8	39	0	1	0	0	394	38	0	0	0	0	0	0	0	0	0	0	0
A8	0	12	0	1	2	3	54	400	2	0	0	0	0	0	0	0	0	0	6
A9	0	0	0	0	13	3	0	0	428	26	9	0	1	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	414	66	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	61	419	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	1	0	463	16	0	0	0	0	0
A14	0	0	0	0	0	0	0	1	1	0	0	0	95	383	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	477	0	1	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	478	0
A19	0	0	0	0	0	0	0	7	0	0	0	1	2	0	0	0	0	0	470

(a)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	479	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	460	20	0	0	0	0	0	0	0	0	0	0	0
A8	0	3	0	0	1	2	34	438	0	0	0	0	0	0	0	0	0	0	2
A9	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	2	478	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	1	0	478	1	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	4	0	0	0	1	0	0	0	0	0	0	475

(b)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	467	13	0	0	0	0	0	0	0	0	0	0	0
A8	0	4	0	0	1	1	43	429	1	0	0	0	0	0	0	0	0	0	1
A9	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	2	478	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	477

(c)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	468	3	3	0	0	0	4	2	0	0	0	0	0	0	0	0	0	0	0
A2	3	455	0	0	0	0	14	8	0	0	0	0	0	0	0	0	0	0	0
A3	4	2	467	1	0	0	3	1	0	0	0	0	0	0	1	0	0	0	1
A4	2	1	1	471	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0
A5	2	0	0	0	466	1	0	5	2	1	1	0	1	0	0	1	0	0	0
A6	1	0	0	0	1	463	1	10	3	0	0	0	1	0	0	0	0	0	0
A7	6	16	1	1	0	0	413	43	0	0	0	0	0	0	0	0	0	0	0
A8	7	8	1	1	6	7	57	372	5	1	1	0	3	1	0	0	1	0	9
A9	3	0	0	0	3	5	1	4	451	4	3	0	2	1	0	1	0	1	1
A10	3	0	0	0	1	0	0	1	3	455	13	0	1	0	0	1	0	0	2
A11	3	0	0	0	0	0	0	1	3	11	456	0	2	2	0	1	0	0	1
A12	1	0	0	0	0	0	0	0	0	1	0	477	0	0	0	0	0	0	1
A13	4	0	0	0	1	1	0	3	2	1	1	0	450	12	1	2	0	0	2
A14	4	0	0	0	0	0	0	1	1	0	2	0	10	458	1	1	0	0	2
A15	2	0	0	0	0	0	0	1	0	0	1	0	0	1	472	3	0	0	0
A16	4	0	0	0	1	0	0	1	1	1	1	0	2	2	5	460	0	1	1
A17	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	477	0	0
A18	2	0	0	0	0	1	0	2	1	0	0	1	0	0	0	0	0	472	1
A19	3	0	0	0	0	1	0	11	1	1	1	1	2	1	0	1	0	1	456

(d)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	462	2	9	0	0	0	4	2	0	0	0	0	0	0	0	0	1	0	0
A2	1	451	0	1	0	1	19	7	0	0	0	0	0	0	0	0	0	0	0
A3	7	0	471	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
A4	1	2	1	471	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	463	0	0	6	2	0	2	0	1	1	0	5	0	0	0
A6	0	0	0	0	0	468	0	9	1	0	0	0	0	0	0	0	0	2	0
A7	5	23	0	1	0	1	399	51	0	0	0	0	0	0	0	0	0	0	0
A8	2	5	0	0	10	14	60	366	3	1	1	0	4	2	1	0	0	1	10
A9	0	0	0	0	3	2	0	3	452	9	6	0	2	0	0	1	0	2	0
A10	0	0	0	0	1	0	0	2	6	452	15	0	1	1	0	1	0	1	0
A11	0	0	0	0	0	0	0	1	4	19	452	0	1	1	1	0	0	0	1
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	1	0	0	5	2	1	1	0	443	22	2	2	0	0	1
A14	0	0	0	0	1	0	0	1	1	1	2	1	23	445	0	4	0	0	1
A15	0	0	0	0	0	0	0	2	0	0	0	0	3	1	469	5	0	0	0
A16	0	0	0	0	5	0	0	1	2	2	2	0	4	4	5	455	0	0	0
A17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	479	0	0
A18	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	476	1
A19	0	0	0	0	0	0	0	12	0	0	0	0	3	0	0	1	0	1	463

(e)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	479	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A3	1	0	479	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	479	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	478	0	2	0	0	0	0	0	0	0	0	0	0	0
A7	0	8	0	0	0	0	457	15	0	0	0	0	0	0	0	0	0	0	0
A8	0	3	0	0	3	4	52	413	1	0	0	0	0	0	0	0	0	0	4
A9	0	0	0	0	0	3	0	0	475	1	1	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	1	477	2	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	2	477	0	0	0	0	0	0	0	1
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	476	4	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	1	479	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	477	2	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	479	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	7	0	0	0	0	1	0	0	0	0	0	472

(f)

Table 4.6: Confusion matrices for (a) NB (93.7%), (b) ANN (99.2%), (c) SVM (99.2%), (d) NB-T (94.9%), (e) J48-T (94.5%), (f) RF-T (98.6%) classifier in WEKA for 10-fold cross validation.

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	467	2	9	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
A2	0	475	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0
A3	0	1	473	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	2
A4	0	0	0	479	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	475	0	5	0	0	0	0	0	0	0	0	0	0	0
A7	4	32	0	0	2	0	407	35	0	0	0	0	0	0	0	0	0	0	0
A8	0	6	0	0	1	4	55	403	4	0	0	0	0	0	0	0	0	0	7
A9	0	0	0	0	4	2	0	2	454	7	7	0	4	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	1	447	32	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	1	22	457	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	2	1	0	0	0	468	9	0	0	0	0	0
A14	0	0	0	0	0	0	0	2	0	0	0	0	8	470	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	479	0	1	0	0
A16	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	477	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	478	0
A19	0	0	0	0	0	0	0	7	0	0	1	0	3	0	0	0	0	0	469

(a)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	445	1	33	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A2	1	463	0	0	0	1	11	4	0	0	0	0	0	0	0	0	0	0	0
A3	2	0	478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	479	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	471	0	7	2	0	0	0	0	0	0	0	0	0	0
A7	13	53	0	1	4	3	383	23	0	0	0	0	0	0	0	0	0	0	0
A8	0	15	0	0	7	11	55	381	5	0	0	0	1	0	0	0	0	0	5
A9	0	0	0	0	13	3	0	0	426	28	6	0	3	1	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	3	417	60	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	1	58	419	0	1	1	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	1	0	471	8	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	11	469	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	476	4	0	0	0
A16	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2	475	0	1	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	477	0
A19	0	0	0	0	0	0	0	9	0	0	0	0	1	0	0	0	0	0	470

(b)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	478	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	477	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	478	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	478	0	2	0	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	467	13	0	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	44	434	0	0	0	0	0	0	0	0	0	0	2
A9	0	0	0	0	0	0	0	1	479	0	0	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	478	2	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0
A14	0	0	0	0	0	0	0	2	0	0	0	0	0	478	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	476

(c)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	477	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	477	0	0	2	0	0	0	0	0	0	0	0	0	0	1
A6	0	0	0	0	0	478	0	1	0	0	0	0	0	0	0	0	0	0	1
A7	0	0	0	0	0	0	468	12	0	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	57	421	0	0	0	0	0	0	0	0	0	0	2
A9	0	0	0	0	0	0	0	2	476	0	0	0	0	0	0	0	0	0	2
A10	0	0	0	0	0	0	0	0	0	478	1	0	0	0	0	0	0	0	1
A11	0	0	0	0	0	0	0	0	0	0	479	0	0	0	0	0	0	0	1
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	1	0	0	0	0	477	1	0	0	0	0	1
A14	0	0	0	0	0	0	0	1	0	0	0	0	0	478	0	0	0	0	1
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	479	0	0	1
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	478

(d)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	477	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	477	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	475	0	0	4	0	0	0	0	0	0	0	0	0	0	1
A6	0	0	0	0	0	476	0	3	0	0	0	0	0	0	0	0	0	0	1
A7	0	0	0	0	0	0	461	19	0	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	47	431	0	0	0	0	0	0	0	0	0	0	2
A9	0	0	0	0	0	0	0	2	477	0	0	0	0	0	0	0	0	0	1
A10	0	0	0	0	0	0	0	1	0	477	1	0	0	0	0	0	0	0	1
A11	0	0	0	0	0	0	0	0	0	0	479	0	0	0	0	0	0	0	1
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	1	0	0	0	0	476	1	0	0	0	0	2
A14	0	0	0	0	0	0	0	1	0	0	0	0	0	478	0	0	0	0	1
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	478

(e)

	classified																		
true	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	479	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A2	0	476	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	479	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	478	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	474	0	0	5	0	0	0	0	0	0	0	0	0	0	1
A6	0	0	0	0	0	476	0	2	0	0	0	0	0	0	0	0	0	0	2
A7	0	0	0	0	0	0	454	25	0	0	0	0	0	0	0	0	0	0	1
A8	0	0	0	0	0	0	46	430	0	0	0	0	0	0	0	0	0	0	4
A9	0	0	0	0	0	0	0	2	477	0	0	0	0	0	0	0	0	0	1
A10	0	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	1	0	0	478	0	0	0	0	0	0	0	1
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	2	0	0	0	0	472	4	0	0	0	0	2
A14	0	0	0	0	0	0	0	1	0	0	0	0	0	477	0	0	0	0	2
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	478

(f)

Table 4.7: Confusion matrices for (a) NB (96.6%), (b) DBC (94.8%), (c) GMM_1 (99.1%), (d) GMM_2 (99.0%), (e) GMM_3 (98.9%), (f) GMM_4 (98.8%) classifier in PRTools for 10-fold cross validation.

	classified	
true	correct	incorrect
A1	441	39
A2	435	45
A3	466	14
A4	463	17
A5	474	6
A6	470	10
A7	346	134
A8	368	112
A9	414	66
A10	422	58
A11	435	45
A12	462	18
A13	455	25
A14	476	4
A15	470	10
A16	465	15
A17	474	6
A18	453	27
A19	447	33

Table 4.8: Number of correctly and incorrectly classified motions out of 480 for ANN classifier in PRTools (10-fold cross validation, 92.5%).

	classification techniques		activities																	
			A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**
WEKA	NB		a	a	g	e	g	g	a	a	a	a	a	e	a	p	e	g	e	g	g
	ANN		e	e	g	e	e	g	g	a	g	g	g	e	g	e	e	e	e	e	g
	SVM		e	e	e	e	e	e	g	a	e	e	g	e	e	e	e	e	e	e	g
	tree methods	NB-T	g	a	g	g	g	g	a	p	a	a	a	g	a	a	g	g	g	g	a
		J48-T	g	a	g	g	g	g	p	p	a	a	a	e	a	a	g	a	g	g	g
		RF-T	g	g	g	e	g	g	a	a	g	g	g	e	g	g	g	g	e	e	g
PRTools	NB		g	g	g	g	g	g	a	a	a	a	a	e	g	g	g	g	e	g	g
	ANN		a	a	g	a	g	g	p	p	a	a	a	g	g	g	g	g	g	a	a
	DBC		e	g	g	g	g	g	g	g	a	g	e	e	g	g	e	e	e	e	g
	GMM	GMM_1	e	g	g	g	g	g	g	a	g	g	e	e	g	g	e	e	e	e	g
		GMM_2	e	g	g	g	g	g	g	a	g	g	g	e	g	g	e	e	e	e	g
		GMM_3	e	g	g	g	g	g	g	a	g	g	g	e	g	g	e	g	e	e	g
		GMM_4	g	g	g	g	g	g	a	a	g	g	g	e	g	g	e	e	e	e	g

Table 4.9: The performances of classification techniques for distinguishing different activity types (categorized as poor (p), average (a), good (g), and excellent (e)). These results are deduced from confusion matrices given in Tables 4.6 and 4.7 according to the number of feature vectors of a certain activity that the classifier correctly classifies [poor (<400), average (in the range 400–459), good (in the range 460–479), excellent (exactly 480)].

4.4 Comparison of Machine Learning Environments

In comparing the two machine learning environments used in this study, algorithms implemented in WEKA appear to be more robust to parameter changes than PRTools. In addition, WEKA is easier to work with because of its graphical user interface (GUI). The interface displays detailed descriptions of the algorithms along with their references and parameters when needed. On the other hand, PRTools does not have a GUI and the descriptions of the algorithms given in the references are insufficient. However, PRTools is more compatible with MATLAB. All in all, both software environments are very strong tools to be used in pattern recognition.

The implementations of the same algorithm in WEKA and PRTools may not be exactly the same. For instance, this is reflected by the difference in correct differentiation rates obtained with NB and ANN classifiers. The higher rates are achieved with NB implemented in PRTools because the distribution of each feature is estimated using histograms. On the other hand, WEKA uses a normal distribution to estimate probability density functions. Considering the ANN classifier, PRTools does not allow the user to set values for the learning and momentum constants which play a crucial role in updating the connection weights. Therefore, the ANN implemented in PRTools is quite incompetent compared with the one implemented in WEKA.

4.5 Previous Results

The results previously reported by our research group indicate that the best method, given its high correct classification rate, relatively small pre-processing and classification times, and storage requirements, is Bayesian decision making

(BDM) [57]. In 10-fold cross-validation scheme, it has a rate of 99.2% which slightly outperforms the GMM_1 classifier with 0.1%. The rates obtained by using ANN and SVM presented in this study are higher than the ones reported in [57]. Especially for L1O cross validation, the differences between the rates are around 3%. These differences arise both from the implementation of the algorithms and the variation in the distribution of the feature vectors in the partitions obtained using RRSS and 10-fold cross-validation schemes. The high rates provided by BDM and GMM_1 in these studies illustrate the high estimation efficiency of multi-variate Gaussian models for activity recognition tasks. However, multi-variate Gaussian models are not well-suited to situations where subject-based L1O cross validation is employed. In such occasions where high generalization accuracy is required, they need to be replaced with ANNs or SVM.

Although both of them are based on the Bayesian approach, there is considerable difference between NB and BDM because of the independence assumption embraced in the NB classifier. The average correct classification rates previously reported by our group for BDM using RRSS and 10-fold cross-validation techniques are 99.1% and 99.2%, respectively [57], whereas these rates drop to 96.5% and 96.6% for NB in this study (Table 4.3). In L1O cross validation however, 75.8% rate achieved in BDM [57] is outperformed by NB with the rate 89.2% (Table 4.3). It can be concluded that independent feature assumption works better for subject-wise partitioning scheme.

4.6 Computational Considerations

The performances of the software environments and implemented classifiers are compared in terms of their execution times. The master software MATLAB is run on a computer with Pentium(R) Dual-Core CPU E520 at clock frequency of 2.50 GHz, 2.00 GB of RAM, and operated with Microsoft Windows XP Home

Edition. Execution times for training and test steps corresponding to all classifiers and both environments are provided in Table 4.10. These times are based on the time it takes for the full L1O cross-validation cycle to be completed. In other words, each classifier is run 8 times for all subjects and the total time of the complete cycle for each classifier is recorded. Assuming that these classification algorithms are used in a real-time system, it is desirable to keep the test times at a minimum.

Considering WEKA, test times are misleading because apart from the time consumed for calculating the correct differentiation rate, several other performance criteria, such as various error parameters and confusion matrices, are calculated during the test step and the time consumed for those calculations are added to the total time given in Table 4.10. The actual test times should be much shorter. In contrast, times concerning PRTools are quite consistent. Therefore, it is not possible to compare these two environments in terms of their classification speed.

Among the decision-tree methods, because they train a NB classifier for every leaf node, NB-T have the longest training time whereas J48-T have the shortest. There is hardly any difference between the test times of decision trees. Therefore, taking its high correct classification rate into consideration, RF-T seems to be the best decision-tree method. In WEKA particularly, tree methods perform better than every other classifier in terms of test times.

In terms of the correct differentiation rates, ANN implemented in WEKA and SVM are superior to the other classifiers. However, training and testing the ANN and SVM takes much longer time than all other techniques but considering the rates obtained using these techniques for all cross-validation schemes, they are the best techniques to be used. SVM classifier takes the most testing time because it uses a Gaussian kernel for mapping and considers every possible pairwise combination of classes during testing phase therefore, it seems better to

prefer ANN instead of SVM. On the other hand, GMM_1 with its short training and test time requirements could be considered except for the L1O case. Considering L1O cross validation, the immediate choice would be the ANN or SVM technique. DBC with its moderate correct classification rates and longest test time is not to be preferred among the classifiers used in PRTools.

	classification techniques		**times (sec)**	
			training	**test**
WEKA	NB		1.66	20.44
	ANN		2416.00	4.50
	SVM		355.32	2356.97
	tree methods	NB-T	2610.90	2.65
		J48-T	24.09	2.65
		RF-T	57.47	2.80
PRTools	NB		0.68	0.48
	ANN		547.77	0.44
	DBC		98.55	1.41
	GMM	GMM_1	1.33	0.46
		GMM_2	161.70	0.58
		GMM_3	129.44	0.72
		GMM_4	118.02	1.06

Table 4.10: Execution times of training and test steps for all classification techniques based on the full cycle of L1O cross-validation method and both environments.

Chapter 5

Conclusion and Future Work

In this chapter, the main inferences drawn from this study are presented, the potential application areas are mentioned, and possible future research directions to be explored are discussed.

In this thesis, we presented the results of a comparative study where features extracted from miniature inertial sensor and magnetometer signals are used for classifying human activities. We compared a number of classification techniques based on the same data set in terms of their correct differentiation rates, confusion matrices, and computational costs. The algorithms of the classification techniques compared are provided on two commonly used open source environments: WEKA and PRTools. The functionality and the manageability of these two environments are also discussed.

In general, the ANN and SVM techniques implemented in WEKA are the best choices in terms of classification performance; however, the computational cost of these methods is very high. The rates achieved by the GMM_1 technique are very close to ANN and SVM classifiers except for the L1O cross-validation scheme and this technique requires quite small computational time compared to other classification techniques. Thus, the GMM technique also seems to be a

suitable method for activity classification problems where it is appropriate to model the feature space with multi-variate Gaussian distributions.

The magnetometer turns out to be the best type of sensor to be used in classification whereas gyroscope is the least useful compared to the other sensor types. However, it should be kept in mind that the absolute information that magnetometer signals provide can be distorted by metal surfaces and magnetized objects in the vicinity of the sensor. Considering location of the sensor units on body, the sensors worn on the legs seem to provide most valuable information on activities.

We implemented and compared a number of different cross-validation techniques in this study. The correct classification rates obtained by subject-based L1O cross validation are usually lower whereas the ones obtained by 10-fold cross validation are usually the highest. Considering the satisfying correct differentiation rates obtained with it, RRSS cross-validation technique has a disadvantage that some feature vectors may never be used for testing, whereas other may be used more than once. In 10-fold and L1O cross validation, all feature vectors are used equally for both training and testing, and each feature vector is used for testing exactly once.

To the best of our knowledge, positioning, number, and type of sensors has not been much studied in the area of activity recognition. Typically, some configuration, number, and modality of sensors is chosen and used without strong justification. The comparative analysis provided in this thesis may guide the researchers working in activity classification using body-worn sensors signals.

There are diverse applications in which the human activity monitoring and classification techniques presented here can be utilized. An essential area of interest could be home-based care and rehabilitation of elderly people; emergency situations such as falls or changes in vital signs could be detected almost instantly.

Proper performance of daily physical therapy exercises assigned, for example, after a surgery can be remotely monitored and feedback can be provided. Similarly, remote monitoring of people with physical or mental disabilities, and children at home, school, or in the neighborhood could be done.

Another potential area that is related to health is medical diagnosis where these techniques can be employed to diagnose a patient and proper treatment can be immediately applied. In order to achieve this, some additional sensors could be necessary.

Similarly, these techniques can be used in the area of physical education, training and sports, and dance to guide the individual to improve his/her skills and prevent injury. In animation and film making, the sensors used in this study can be used in complementary fashion with cameras to develop realistic animated models.

In entertainment, video games could be much more realistic and appealing with wearable inertial sensors integrated into the game in which these classification techniques are embedded for recognizing the moves made by the player. There are some simple games with this capability however rather complex moves could be possible in the game if these techniques are used.

Today's advanced cell phone industry has many opportunities as well. These devices are carried by almost everyone during the day. It would be quite beneficial to develop an application for an iPhone to monitor the daily activities that are performed by the individual. An iPhone seems to be the best candidate since it has an embedded 3D inertial sensors (gyroscopes and accelerometers) and developing an application for this device is quite common. It is possible to perform classification of activities instantly, once the individual is instructed on where to locate the device on the body.

There are several future research problems to investigate in activity recognition and classification:

An activity recognition system should be able to recognize and classify as many activities as possible while maintaining the performance already achieved; therefore, a broader activity spectrum is a necessity. In addition, a further set of unclassified activities should be taken into account to prevent the system from making an incorrect decision and that requires a separate class of unexpected activities to be defined to the classifiers. In order to define such a class of activities, each activity that is already defined to the system needs to be normalized and should have certain boundaries in the feature space. However, this requires an extensive study because, for example, different individuals perform the same activities in a different way.

Fall detection and classification is another research area that has not been sufficiently well investigated [22], due to the difficulty of performing realistic experiments in this area [13]. There is no standard definition of falls and a systematic technique for detecting them does not exist. As the average age of population increases, it seems vital to develop such definitions and techniques as soon as possible [20].

An aspect that could be further investigated is the sensor-feature relevance. In most of the activity recognition studies, it is desired to feed the classifiers with the most informative and discriminative features. Especially, in the case of fusing several types of sensors, each sensor type could be associated with particular set of features that are the most discriminative in terms of classifying activities.

Considering the sensors used in these systems, all of the potential applications mentioned and our study itself suggest the need for a systematic framework for optimizing the number, positioning, and type of sensors used in activity

recognition. Otherwise, it is quite difficult to apply the techniques that are proposed to a real-time system.

Bibliography

[1] Xsens Technologies B.V., Enschede, Holland, *MTi and MTx User Manual and Technical Documentation*, 2009. `http://www.xsens.com`.

[2] I. J. Cox and G. T. Wilfong, ed., *Autonomous Robot Vehicles*. New York: Springer-Verlag, 1990. Section on Inertial Navigation edited by M. M. Kuritsky, and M. S. Goldstein.

[3] D. A. Mackenzie, *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*. Cambridge, MA: MIT Press, 1990.

[4] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 328–342, June 1995.

[5] B. Barshan and H. F. Durrant-Whyte, "Evaluation of a solid-state gyroscope for robotics applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 44, pp. 61–67, February 1995.

[6] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, pp. 2520–2530, December 2005.

[7] J. G. Nichol, S. P. N. Singh, K. J. Waldron, L. R. Palmer, III, and D. E. Orin, "System design of a quadrupedal galloping machine," *International Journal of Robotics Research*, vol. 23, pp. 1013–1027, October-November 2004.

[8] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, pp. 932–943, October 2006.

[9] W. T. Ang, P. K. Khosla, and C. N. Riviere, "Design of all-accelerometer inertial measurement unit for tremor sensing in hand-held microsurgical instrument," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, (Tapei, Taiwan), pp. 1781–1786, 14–19 September 2003.

[10] W. T. Ang, P. K. Pradeep, and C. N. Riviere, "Active tremor compensation in microsurgery," in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, (San Francisco, CA), pp. 2738–2741, 1–5 September 2004.

[11] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. U.K.: IEE, 2nd ed., 2004.

[12] W. Zijlstra and K. Aminian, "Mobility assessment in older people: new possibilities and challenges," *European Journal of Ageing*, vol. 4, pp. 3–12, March 2007.

[13] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, pp. R1–R20, April 2004.

[14] W. Y. Wong, M. S. Wong, and K. H. Lo, "Clinical applications of sensors for human posture and movement analysis: A review," *Prosthetics and Orthotics International*, vol. 31, pp. 62–75, March 2007.

[15] A. M. Sabatini, *Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation*, pp. 70–100.

Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques, Idea Group Publishing, 2006.

[16] O. Tunçel, K. Altun, and B. Barshan, "Classifying human leg motions with uniaxial piezoelectric gyroscopes," *Sensors*, vol. 9, pp. 8508–8546, October 2009.

[17] F. Audigié, P. Pourcelot, C. Degueurce, D. Geiger, and J. M. Denoix, "Fourier analysis of trunk displacements: a method to identify the lame limb in trotting horses," *Journal of Biomechanics*, vol. 35, pp. 1173–1182, September 2002.

[18] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 119–128, January 2006.

[19] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. C. F. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, pp. 679–687, September 2004.

[20] K. Hauer, S. E. Lamb, E. C. Jorstad, C. Todd, and C. Becker, "Systematic review of definitions and methods of measuring falls in randomised controlled fall prevention trials," *Age and Ageing*, vol. 35, pp. 5–10, January 2006.

[21] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection—principles and methods," in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1663–1666, August 2007.

[22] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, pp. 285–291, August 2008.

[23] W. H. Wu, A. A. T. Bui, M. A. Batalin, D. Liu, and W. J. Kaiser, "Incremental diagnosis method for intelligent wearable sensor system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, pp. 553–562, September 2007.

[24] E. Jovanov, A. Milenkovic, C. Otto, and P. C de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 6, 2005.

[25] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 20–26, January 2008.

[26] R. Aylward and J. A. Paradiso, "Sensemble: A wireless, compact, multi-user sensor system for interactive dance," in *Proceedings of the Conference on New Interfaces for Musical Expression*, (Paris, France), pp. 134–139, 4–8 June 2006.

[27] J. Lee and I. Ha, "Real-time motion capture for a human body using accelerometers," *Robotica*, vol. 19, pp. 601–610, September 2001.

[28] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, vol. 27, December 2008.

[29] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, March 2001.

[30] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, pp. 90–126, November-December 2006.

[31] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, pp. 585–601, March 2003.

[32] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, vol. 73, pp. 428–440, March 1999.

[33] U. Töreyin, Y. Dedeoğlu and E. Çetin, "HMM based falling person detection using both audio and video," in *IEEE International Workshop on Human Computer Interaction*, vol. 3766, pp. 211–220, Berlin, Heidelberg: Springer-Verlag, 2005. Lecture Notes in Computer Science.

[34] L. Hyeon-Kyu and J. H. Kim, "An HMM-based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 961–973, October 1999.

[35] H. Junker, O. Amft, P. Lukowicz, and G. Troester, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, pp. 2010–2024, June 2008.

[36] J. C. Lementec and P. Bajcsy, "Recognition of arm gestures using multiple orientation sensors: gesture classification," in *Proceedings of the 7th International Conference on Intelligent Transportation Systems*, pp. 965–970, Washington D.C., U.S.A., 3–6 October 2004.

[37] M. Uiterwaal, E. B. C. Glerum, H. J. Busser, and R. C. van Lummel, "Ambulatory monitoring of physical activity in working situations, a validation study," *Journal of Medical Engineering and Technology*, vol. 22, pp. 168–172, July/August 1998.

[38] J. B. Bussmann, P. J. Reuvekamp, P. H. Veltink, W. L. Martens, and H. J. Stam, "Validity and reliability of measurements obtained with an "activity monitor" in people with and without transtibial amputation," *Physical Therapy*, vol. 78, pp. 989–998, September 1998.

[39] K. Aminian, P. Robert, E. E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," *Medical and Biological Engineering and Computing*, vol. 37, pp. 304–308, May 1999.

[40] D. Roetenberg, P. J. Slycke, and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 883–890, May 2007.

[41] B. Najafi, K. Aminian, F. Loew, Y. Blanc, and P. Robert, "Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly," *IEEE Transactions on Biomedical Engineering*, vol. 49, pp. 843–851, August 2002.

[42] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Büla, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 711–723, June 2003.

[43] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation," *International Journal of Robotics Research*, vol. 26, pp. 607–624, June 2007.

[44] T. Viéville and O. D. Faugeras, *Cooperation of the Inertial and Visual Systems*, vol. F63 of *NATO ASI Series*, pp. 339–350. Berlin, Heidelberg, Germany: Springer-Verlag, 59th ed., 1990. in Traditional and Non-Traditional Robotic Sensors.

[45] in *Proceedings of the Workshop on Integration of Vision and Inertial Sensors (InerVis)*, Coimbra, Portugal, June 2003; Barcelona, Spain, April 2005.

[46] Special Issue on the 2nd Workshop on Integration of Vision and Inertial Sensors (InerVis05) *International Journal of Robotics Research*, vol. 26, June 2007.

[47] R. Zhu and Z. Zhou, "A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, pp. 295–302, June 2004.

[48] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2526–2533, Rome, Italy, 10–14 April 2007.

[49] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of Pervasive Computing, Lecture Notes in Computer Science*, vol. 3001, pp. 1–17, 2004. `web.media.mit.edu/~intille/papers-files/BaoIntille04.pdf`.

[50] P. H. Veltink, H. B. J. Bussmann, W. de Vries, W. L. J. Martens, and R. C. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, pp. 375–385, December 1996.

[51] K. Kiani, C. J. Snijders, and E. S. Gelsema, "Computerized analysis of daily life motor activity for ambulatory monitoring," *Technology and Health Care*, vol. 5, pp. 307–318, October 1997.

[52] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, pp. 571–583, September 1999.

[53] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 156–167, January 2006.

[54] F. R. Allen, E. Ambikairajah, N. H. Lovell, and B. G. Celler, "Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models," *Physiological Measurement*, vol. 27, pp. 935–951, October 2006.

[55] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, pp. 1166–1172, September 2010.

[56] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Proceedings of European Symposium on Ambient Intelligence, Lecture Notes in Computer Science*, vol. 2875, (Eindhoven, The Netherlands), pp. 220–232, November 2003. `citeseer.ist.psu.edu/kern03multisensor.html`.

[57] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, October 2010.

[58] M. Lustrek and B. Kaluza, "Fall detection and activity recognition with machine learning," *Informatica*, vol. 33, pp. 205–212, May 2009.

[59] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, U.S.A.: Morgan Kaufmann, Elsevier, 2nd ed., 2005.

[60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 10–18, June 2009. `http://www.cs.waikato.ac.nz/ml/weka/`.

[61] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. M. J. Tax, and S. Verzakov, *PRTools4.1 a MATLAB Toolbox for Pattern*

Recognition. Delft University of Technology, 2600 AA, Delft, The Netherlands, August 2007. `http://www.prtools.org/download.html`.

[62] M. C. Yüksek and B. Barshan, "Human activity classification with miniature inertial and magnetic sensor signals," in *Proceedings of 19th European Signal Processing Conference*, Barcelona, Spain, 29 August–2 September 2011.

[63] M. C. Yüksek and B. Barshan, "Human activity classification with miniature inertial and magnetic sensors," in *Proceedings of 19th Signal Processing and Communications Applications (SIU) Conference*, Antalya, Turkey, 20–22 April 2011.

[64] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 871–879, March 2009.

[65] A. Webb, *Statistical Pattern Recognition*. New York: John Wiley & Sons, 2002.

[66] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of IEEE International Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, Montreal, Quebec, Canada, 18–20 August 1995.

[67] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, Inc., 2001. 2nd edition.

[68] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*. Boston, MA: PWS Publishing, 1996.

[69] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan Publishing, 1994.

[70] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN, U.S.A.: West Publishing, 1992.

[71] E. Pekalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recognition*. River Edge, NJ, U.S.A.: World Scientific Publishing Co., 2005.

[72] E. Pekalska and R. P. W. Duin, "Dissimilarity-based classification for vectorial representations," in *Proceedings of 18th International Conference on Pattern Recognition*, (Hong Kong), pp. 137–140, 20–24 August 2006.

[73] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid," in *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, pp. 202–207, Portland, OR, U.S.A., 2–4 August 1996.

[74] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, U.S.A.: Morgan Kaufmann Publishers, 1993.

[75] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[76] R. Kohavi and J. R. Quinlan, "Decision tree discovery," in *Handbook of Data Mining and Knowledge Discovery*, pp. 267–276, University Press, 1999.

[77] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, January 2001.

[78] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep. TR-97-021, University of Berkeley, International Computer Science Institute, Berkeley, CA, U.S.A., April 1998. `citeseer.ist.psu.edu/bilmes98gentle.html`.

[79] C. W. Hsu, C. C. Chang, and C. J. Lin, *A Practical Guide to Support Vector Classification*, 2008.

[80] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, November 2001.

[81] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, June 1998.

[82] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Introduction to Support Vector Learning*. Cambridge, MA, U.S.A.: MIT Press, 1999.

[83] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *The Annals of Statistics*, pp. 507–513, MIT Press, 1996.

[84] J. C. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," Tech. Rep. MSR-TR-98-14, Microsoft Research, Redmond, WA, April 1998.

[85] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," *Informatica*, vol. 31, pp. 249–268, October 2007.