Pers Ubiquit Comput (2013) 17:675–695
DOI 10.1007/s00779-012-0515-4

ORIGINAL ARTICLE

Activity logging using lightweight classification techniques in mobile devices

Henar Martín · Ana M. Bernardos · Josué Iglesias · José R. Casar

Received: 1 October 2011 / Accepted: 1 February 2012 / Published online: 5 April 2012

Abstract Automated activity recognition enables a wide variety of applications related to child and elderly care, disease diagnosis and treatment, personal health or sports training, for which it is key to seamlessly determine and log the user's motion. This work focuses on exploring the use of smartphones to perform activity recognition without interfering in the user's lifestyle. Thus, we study how to build an activity recognition system to be continuously executed in a mobile device in background mode. The system relies on device's sensing, processing and storing capabilities to estimate significant movements/postures (walking at different paces—slow, normal, rush, running, sitting, standing). In order to evaluate the combinations of sensors, features and algorithms, an activity dataset of 16 individuals has been gathered. The performance of a set of lightweight classifiers (Naïve Bayes, Decision Table and Decision Tree) working on different sensor data has been fully evaluated and optimized in terms of accuracy, computational cost and memory fingerprint. Results have pointed out that a priori information on the relative position of the mobile device with respect to the user's body enhances the estimation accuracy. Results show that computational low-cost Decision Tables using the best set of features among mean and variance and considering all the sensors (acceleration, gravity, linear acceleration, magnetometer, gyroscope) may be enough to get an activity estimation accuracy of around 88 % (78 % is the accuracy of the Naïve Bayes algorithm with the same characteristics used as a baseline). To demonstrate its applicability, the activity recognition system has been used to enable a mobile application to promote active lifestyles.

Keywords Activity recognition · Life logging · Pattern recognition · Context awareness · Mobile application · Personal health

1 Introduction

It is estimated that, nowadays, nearly 40 % of the mobile telephones in the Western European and US markets are smartphones; in Asia Pacific, the penetration share is about 20 %, but rapidly increasing [1, 2]. Smartphones' technological capabilities have improved in the last few years, significantly superseding 'dumb phones,' in terms of memory, computational power, sensors and connectivity features [3]. The rising memory and processing power of these phones allow the enhancement of embedded calculations and local information storage, enabling the mobile to work off-line, thus contributing to save battery and reducing security problems related to communications. The availability of additional communication interfaces (such as WiFi, Bluetooth, NFC or ZigBee) facilitates, for example, extended localization features, connectivity to external devices and networking capabilities. With respect to their sensing features, smartphones may be considered as sensing platforms: they include proximity and light sensors, front and back cameras and microphones, and inertial systems (accelerometers, gyroscopes) as well as digital compasses.

H. Martín (✉) · A. M. Bernardos · J. Iglesias · J. R. Casar
Telecommunications School, Universidad Politécnica de Madrid, Av. Complutense 30, 28040 Madrid, Spain
e-mail: hmartin@grpss.ssr.upm.es

A. M. Bernardos
e-mail: abernardos@grpss.ssr.upm.es

J. Iglesias
e-mail: josue@grpss.ssr.upm.es

J. R. Casar
e-mail: jramon@grpss.ssr.upm.es

Due to their new functionalities, which go beyond the traditional phone calls and texting, these devices have become everyday devices that many people carry with them all day long [4, 5]. As a consequence, smartphones may facilitate acquiring, logging and even processing personal, environmental or social data, captured through virtual and/or in-device physical sensors.

Hidden in mobile data, there is a huge amount of information that may enable diverse life-logging applications. In particular, many of these applications (refer to Sect. 2.1 for a review) will rely on activity records. What the user is doing at a given moment may be inferred through different strategies. The simplest one consists in the user being responsible for manually annotating or labeling the type of activity he is carrying out [6]. Obviously, manual annotation, even if supported by usable interfaces, is a very demanding task for the user. Additionally, it is also prone to human errors, making it necessary to deal with incorrect, incomplete or missing data from the application side (both by applying a posteriori corrective logic and by triggering specific questions to the user to clarify—which increases the number of times that the user has to be interrupted by the application). Thus, the challenge is to design non-intrusive automated solutions for activity recognition, to simplify and make more robust the measurement collection process and mitigate the needs of user interaction.

'Activity recognition' is a term that is usually used in the literature to refer to both physical posture-movement estimation (low-level activity estimation: for example, 'running,' 'cycling') [7–9] and action-attitude estimation (high-level activity estimation: for example, 'eating,' 'watching TV,' 'studying') [10–12]. The latter may be inferred, for example, from interaction with objects [10] or by fusing movement data with additional context data such as location [11, 12] and ambient sound [11]. Basically, low-level activity estimation is a key input for further reasoning on high-level activity inference. Thus, for the rest of the paper, we will assimilate activity recognition to the process of detecting the posture or movement of the user.

When aiming at building ubiquitous applications, ready to work in uncontrolled environments, it is frequent to retrieve activity data from one or more wearable sensors. For daily living, even the most compact solution comes to be bulky, intrusive and difficult to maintain. The fact that people carry and use mobile devices makes these sensing platforms ideal—in terms of user adoption—to gather ubiquitous low-level activity information and, one step further, to get adherence to activity-logging applications. Evidently, the use of these devices as activity data loggers and processors has operational and functional limitations. On the one hand, users may not carry the sensing device at every moment of the day; on the other hand, mobile devices cannot be placed in some body position facilitating activity recognition (e.g., on the feet). Additionally, sensors quality may be worse when compared to a wearable one (see Sect. 3). Nevertheless, the technology and usability features offered by smartphones are good enough to face the design of fully mobile-based activity recognition systems.

To the best of our knowledge, there are still very few works that consider embedded stand-alone physical activity estimation in smartphones. Thus, in this paper, we explore how to extract information on physical activity just by fusing data from a set of built-in mobile sensors, to extract a set of design rules for activity mobile recognition. Our objective is to infer the user's physical activity (with the shortest possible delay and the highest accuracy) by using a light and ubiquitous strategy, ready to be non-stop executed in a smartphone in co-existence with the rest of its software components. In this way, we propose and compare different pattern recognition schemas to infer whether a user is, for example, walking at different paces, running, sitting or standing. As we want our system to respond to the fact that in real operation, the user may place the mobile in different parts of his body, or inside wearable elements (handbags or backpacks), we also study the impact of adding a pre-processing stage to estimate the mobile relative position. All in all, we analyze how to get the best accuracy, how to diminish the computational cost or how to design an algorithm that may be used on a single sensor (accelerometer) to reduce computational load and memory requirements. The proposed strategies are first validated on a dataset composed of samples of 16 individuals. Afterward, to demonstrate the practical applicability of our analysis, an instance of the fusion system is integrated in a mobile application that aims at continuously monitoring and evaluating the activity level of a user, to inform him about potential sedentary habits.

The paper is organized as follows. Section 2 first reviews some life-logging applications enabled by mobile physical activity recognition; secondly, it gathers a survey on the techniques used to recognize activity. Then, Sect. 3 fully details the experimental scenario for analysis. Our embedded architecture for activity recognition is described in Sect. 4, together with an analysis of key design aspects related to sensors, features and classifiers choice. The dataset that we have gathered to train and evaluate the system is introduced in this section as well. Performance results are presented in Sect. 5. From them, we design the activity recognition solution that has been implemented to cover the needs of a real activity-logging-based application: an activity monitor to prevent sedentary behavior. The application is further described and demonstrated in Sect. 6. Section 7 summarizes our conclusions on how to implement a reliable embedded activity recognition tool.

2 State of the art

2.1 Mobile activity logging as application enabler

Mobile physical activity recognition makes possible the design of diverse logging applications. Following we mention some of them, exemplifying with previous works the type of functionalities and application areas that may be addressed: personal health and wellness, sport training, privacy and security, social based applications, personal diaries, etc.

Activity recognition facilitates designing (i) wellness self-monitors [13–16] ready to keep track of the user's lifestyle, in order to provide him with timely feedback to prevent diseases related to sedentary behaviors. Actually, many chronic diseases (asthma, cardiovascular problems, diabetes mellitus, arthritis or mental diseases [17]) get worse under inactive lifestyles, so (ii) disease-specific coaches [18, 19] and activity-based games [20] to enhance treatment adherence in terms of daily movement are being explored as well. Activity recognition systems may also detect abnormalities in ambulation, enabling (iii) early diagnose and subsequent monitoring of neurological diseases, in particular of those implying memory loses [9] (Alzheimer's disease) or psychomotor damages [21] (stroke or Parkinson's disease). Together with emotion inference [13, 22], activity analysis may support (iv) tracking psychological problems such as depression, which usually cause behavioral and social pattern alterations. (v) Sports trainers, recording the user's performance and personalizing training plans on real data are still to come, but there are several analyses in the literature which determine how to identify gym [15] and outdoors sportive activities, and specific sports patterns (e.g., soccer [23]). To date, there have been some initiatives to design (vi) emergency detectors, applications designed to detect and notify abnormal situations which may represent a risk for the supervised user [24], as falling down or getting lost (which may be derived from confusing ambulation). Together with location information and/or interaction logging, activity data may help to build (vi) 'routine learners.' Posterior routine log analysis may support planning and logistics (both at family and at professional level) and human behavior analysis (individual and collective). (vii) Social modelers, which aim at characterizing interactions between individuals and complex social systems, may help to understand how information flows in an organization or predicting the spread of a disease [25]. (viii) Automatic diary builders generate records of the user's life for any personal purpose, even in creative ways: InSense [11] takes photographs in the most relevant moments of the day, and the Affective Diary [26] represents data from the device sensors through abstract body shapes.

Additionally, physical activity recognition may be used to improve the user experience with mobile devices, enabling (ix) permanent identification through continuous gait analysis [27]. This can be used to secure the mobile device itself and services implying the use of personal data (e.g., payment). (x) Dynamic interface adaptation is feasible by taking into account activity: for example, if in the bus, the menu icons might grow bigger to facilitate tapping on them. Finally, activity may (xi) improve the performance of localization systems. [12, 28, 29] are some works fusing activity with position estimates.

2.2 Activity classification techniques for mobile devices

Although activity classification is a well-known research topic, solutions relying on mobile embedded sensors and processors have not been widely explored yet. Below there is a review on recent works that adopt this perspective, targeting different applications or functionalities. In brief, Table 1 summarizes the state-of-the-art. Firstly, all the works rely on the use of accelerometers to identify basic movements, although they differ in the signal features employed for activity classification, varying from raw data [29, 30] to time and frequency-domain features [31–33] or Kernel discriminant analysis applied to autoregressive coefficients and signal magnitude area [34]. With respect to the set of body positions where the device may be placed, some of the analysis let the user choose one position for all the experiments [30], while others determine one [28] or multiple [31, 33, 34] fixed positions to gather training data. Although some of the works analyze the accuracy enhancement when a priori information about the mobile position is introduced (e.g., [31]), none of the works intend to estimate where the user is carrying the mobile device, and the positions considered are limited to pockets in most of the works and a clip belt and lanyard (neck cord) in one of them. On the other hand, Yang et al. [32] computes the vertical and horizontal projection over gravity of the acceleration to reduce the effect of the position on the signals gathered from the accelerometer. In addition, the sets of activities that are included in most of the works are standing, sitting, walking, lying and running [28, 30, 33], together with more complex activities such as driving, bicycling or ascending or descending stairs [31, 32, 34].

Except [28, 33] (which process part of the data inside the device and part in an external server), the other systems gather and process data outside the mobile device, using it as if it were another wearable sensor capable of sampling and storing sensor data. This fact influences the type of algorithms used which are Decision trees and Bayesian in the case of the works that carry out classification in the mobile [28, 33] and rules-based algorithms, support vector machines and artificial neural networks, in the rest of the works. These algorithms require more computational or memory capabilities.

Table 1 Comparison of the characteristics of previous works, all of them based on acceleration data

Work	Ofstad et al. [28]	Khan et al. [34]	Zhang et al. [30]	Sun et al. [31]	Yang [32]	Miluzzo et al. [33]
In-device computation?	Yes	No	No	No	No	Yes
Delay?	Windows 1 min long without overlap	2 s windows with no overlap	Nothing mentioned	1 s sliding windows with 50 % overlap	10 s windows, stored every 4 min	Periodically (not included)
Activities	Standing Sitting	Sitting Walking* Walking-upstairs* Walk-downstairs* Running* * Different speeds	Sitting Standing Lying Walking Posture transition Gentle motion	Stationary Walking Running Bicycling Ascending stairs Descending stairs Driving	Standing Walking Running Driving Bicycling	Walking Running Sitting Standing
Type of device (OS)	Nokia N95 (Symbian)	Samsung TOmnia SCH-M490 (Windows mobile 6)	HTC touch HD (Windows mobile 6)	Nokia N97 (Symbian)	Nokia N95 (Symbian)	Nokia N95 (Symbian)
Classifiers	Bayesian	Artificial Neural Networks (ANN) feed-forward backpropagation algorithm	1. Rule-based reasoning module is used to discriminate between “motion” and “motionless” activities 2. 2 multiclass SVM classifiers are used to determine the motion or motionless activity	SVM based classifier	C4.5 Decision Trees (DT) Naïve Bayes (NB) k-Nearest Neighbor (kNN) Support Vector Machine (LibSVM)	J48 Decision Tree algorithm
Features	Raw data	1. Autoregressive Coefficients 2. Signal Magnitude Area 3. Linear Discriminant Analysis 4. Kernel Discriminant Analysis	Raw data	1. Mean 2. Variance 3. Correlation 4. FFT energy 5. Frequency-domain entropy	1. Mean 2. Standard deviation 3. Zero crossing rate 4. 75 % percentile 5. Interquartile range 6. Power spectrum centroid 7. Frequency-domain entropy 8. Cross-correlation between the horizontal and vertical magnitudes	1. Variance 2. Mean 3. Number of peaks
Mobile position	Right trousers pocket	Shirt’s top pocket, Jeans’ front-left pocket, Jeans’ front-right pocket, Jeans’ rear pocket, coat’s inner pocket	Wear the phone in the same place on their bodies for all the activities, but users could individually choose	6 pockets (2 front, 2 rear trousers pockets and 2 front jacket pockets) 4 positions inside the pockets	Indifferent (vertical and horizontal components computed)	Pocket, on a lanyard, clipped to a belt

Table 1 continued

Work	Ofstad et al. [28]	Khan et al. [34]	Zhang et al. [30]	Sun et al. [31]	Yang [32]	Miluzzo et al. [33]
Sample frequency (Hz)	1	45	1	10 (averaging data)	36	
Database size (no. of subjects)	1	6	10	7	4	8
Accuracy	98.9 % (50–50 cross-validation)	96 %	82.8 % (leave-one-out)	91.6 % (unknown position) 94.8 % (Known position) 10-fold cross-validation	91 % best performance 10-fold cross-validation	68.18 % sitting 78.44 % standing 94.44 % walking 74.51 % running

Regarding the performance of the classifiers, most works [28, 31, 32, 34] claim over 90 % accuracy using cross-validation as the test method. Zhang et al. [30] obtain an accuracy over 80 % using leave-one-out method, and Miluzzo et al. [33] provide the information related to every activity detected, which ranges from 70 to 95 %. The number of individuals to test the results varies in the systems proposed from 1 to 10.

Considering this analysis, it is important to mention that we aim at offering a pool of algorithms for activity inference, to be chosen depending on the application and mobile device's requirements in terms of accuracy, computational load and battery consumption. Unlike the works above, our classifiers profit from the information provided from other sensors apart from the accelerometer; these results are compared with the ones obtained with the accelerometer alone.

In this way, our work provides some guidelines to build lightweight systems for the activity with mobile devices. To enhance its performance, we firstly assume that we can know in advance where the mobile device is placed with respect to a body reference. As this information is not always available, we propose a classifier algorithm to detect the relative position of the mobile phone with respect to the user's body before proceeding to estimate his activity.

3 The smartphone as a tool for activity recognition

3.1 Problem statement

As previously said, the aim of this work is to study how to design an embeddable activity recognition system working on mobile sensor data and performing all the calculations in the device (such as the demo application presented in Sect. 6). These applications may usually require the activity recognition functionality to work in background all day long, so it is important to consider any aspect affecting coexistence with the rest of the software elements in the mobile device (e.g., memory load or battery expenditure). Thus, we are going to explore different ways to improve the accuracy on activity recognition and response performance, while minimizing resource consumption and computation time. This will be possible by combining (i) classification strategies, (ii) features selection and (iii) sensors choice. Classification algorithms have to be as light as possible, so we are going to consider different sets of attributes to find the best trade-off between accuracy and memory load (see Sect. 4). With respect to sensors choice, although we will take advantage of the whole set of sensors embedded in a full-fledge device (inertials, light, proximity) to get the best accuracy, we are going to consider the possibility of relying only on the accelerometer, to reduce the computational load and memory size of the classifier.

Additionally, in an ideal scenario, the activity recognition system should work naturally for the user, so he would not need to place the mobile device in any specific position for the activity recognition to work. To achieve so, it is important to know the impact of the on-body device position in the mobile activity recognition process. Our results show that having a priori information about where the mobile device is placed makes the estimation better, so a classification stage to detect the on-body device position has been added to our activity recognition system. As far as we know, this possibility has not been implemented in practice in previous works.

To make our analysis, we have initially chosen a set of activities that have a direct translation into energy expenditure and that are common in standard daily settings (office, home, commuting, etc.). This initial set of activities is composed of: *Slow*, *Normal* and *Rush walking*, *Running*, *Standing*, and *Sitting*. In Sect. 6, we show how translating activity intensity into relative energy expenditure, for example, allows evaluating, in an aggregated manner, whether the user is developing sedentary habits or exercising enough to fulfill therapeutic or training thresholds.

With respect to on-body device positions, we explore how activity recognition works when the user carry the mobile device: in the hand, texting or talking, in the front and back trouser pockets, in the shirt and jacket pockets, in a short-strap and long-strap bag, in a backpack, in an armband and in a waist case. We consider that these are the most common positions in which a user can carry the mobile, both men and women [35]. Specifically, the shirt pocket and short-strap bag positions have only been considered for men and women, respectively.

3.2 Analysis of mobile sensing technology

When facing activity recognition with mobile sensors, it is relevant to know to which extent the sensing system is offering enough quality. In this work, an Android device (Nexus S from Google) equipped with inertial, proximity and light sensors is used.

Table 2 gathers relevant information on the mobile sensors, about their maximum range, minimum delay between two events, resolution and power consumption while the sensor is in use. The measurements from light and proximity sensors have been calculated on events that occur when the measurement of the sensor changes. The sampling rate of the rest of sensors is limited by the minimum delay shown in Table 2. As no real sampling frequency is stated in the sensors' specifications, the measurements gathered in the different experiments (see Sect. 4.1) have been analyzed to characterize this parameter. It happens that sampling rates are not constant (measurements are not perfectly periodic due to device multitasking), so, for this reason, Table 3 shows the mean value of the time between measurements and the standard deviation of those values.

If we compare the measurements range of the sensors embedded in the smartphone with 'higher-quality' inertial sensors that have been used to recognize activity (e.g., MTx [37]), it can be noted that in the case of the accelerometer, this value is lower (MTX—50 m/s^2). Nevertheless, the smartphone range is enough to fulfill activity classification, due to the fact that the acceleration measured is under 2 g in the positions considered.

Apart from that, the maximum range of the magnetometer is also lower than in other sensors (MTX—7.2 mT), but once again enough to be able to estimate the orientation of the device. With respect to the embedded gyroscope, it presents higher maximum range than wearable sensors (MTX—5.2 rad/s).

Regarding the sampling rate, the one corresponding to the accelerometer and the magnetometer is lower in comparison with MTX (50 Hz), whereas the rate of the gyroscope is higher for the smartphone. In the case of the mobile device, the sample rate can be increased by choosing a different sampling rate from the ones offered in the native (Android) sensors API (normal, fastest, game, user interface), but the one chosen (normal) is enough to detect changes in orientation and movement to recognize activity. This is because the analysis of the step frequency for all the

Table 2 Information of the sensors integrated in Nexus S device [36]

Sensor	Units	Maximum range	Minimum delay (μs)	Power (mA)	Resolution	Vendor
Acceleration gravity linear acceleration	m/s^2	19.6	20	0.23	0.0095768	STMicroelectronics
Magnetic field sensor	μT	2000.0	30	6.8	0.0625	Asahi Kasei Microdevices
Orientation	Degrees (°)	360.0	30	7.8	0.015625	Asahi Kasei Microdevice
Light	Lux	3000.0	0	0.75	1.0	Sharp
Proximity	Binary (near/far)	5.0	0	0.75	5.0	Sharp
Gyro	Radians/s	34.906586	1.2	6.1	0.0012217	STMicroelectronics
Rotation	Quaternion	1.0	20	7.03	5.96047E-8	Google Inc.

Table 3 Analysis of the sensors' real sampling frequency (seconds/frequency)

Acceleration linear acceleration gravity	Mean (s/Hz)	0.16/6.25	Gyro	Mean (s/Hz)	0.01/100
	σ (s)	8.67e-4		σ (s)	8.37e-4
Magnetometer	Mean (s/Hz)	0.13/7.69	Orientation	Mean (s/Hz)	0.09/11.11
	σ (s)	0.001		σ (s)	0.06

users in the dataset shows that, in the worst case, the step frequency is 0.35 s/step, so even with a data rate of 6.25 Hz several samples of each step are gathered, making possible to recognize the movement when several steps are taken into consideration in the same processing window. Figure 3 (in Sect. 4.4) shows the samples of a signal gathered from the z axis of the accelerometer during a slow walking test.

Higher sampling frequency also implies higher power consumption. The time that takes to lower the battery level from 95 to 80 % has been measured using a tool that subscribes to the measurements of all the sensors available and stores them in a file for different sensor sampling rate. Under the same conditions (screen always on, this application running alone, WiFi and Bluetooth enabled), the results are 104 min when the frequency rate is normal and 53.5 min for the fastest sampling rate. Even if those measurements are related to the specific device and we cannot take them as significant in terms of absolute values, we can effectively state that battery consumption increases with frequency rate. Moreover, a higher data rate means that more samples are gathered in each window and the calculation of the features becomes more demanding.

4 Analysis of pattern classification techniques for light activity recognition

4.1 Dataset for training and analysis

In order to gather the data necessary to evaluate the performance of the different classifiers tested for activity recognition, a specific dataset has been created. For this purpose, 16 individuals (12 male and 4 female walkers, between the ages of 23 and 50) have been requested to perform a set of activities, each of them during 1 min, carrying or holding the mobile in a specific body position or in wearable elements. In this way, the dataset stores around 90-min data per user. The *Walking* (at different paces) and *Running* sample data have been gathered using a treadmill, adjusting the speed to the pace of each user (Fig. 1). The treadmill is used due to the fact that previous experiments show that individuals tend to change the pace from slow to normal walking if they are not reminded periodically. In addition, the treadmill measures the distance that has been walked, which is helpful to analyze the specific walking characteristics of each individual.

To facilitate the data gathering process, an application that leads the user through the execution of each test has been developed (Fig. 1). All the activities have been performed for each different position, except for activities that are not realistic (e.g., texting and running).

4.2 System architecture overview

Following our system for activity recognition is described. As the reader will notice, it is based on well-known pattern recognition and classification techniques. In brief, these techniques basically consist in assigning the corresponding class (in our problem, for example, a type of activity) to an instance that is provided as an input. This instance is composed of attributes, information of the sensor signals that characterizes and makes possible to differentiate the

Fig. 1 Execution of some of the tests

classes (e.g., variance of the acceleration measured in the z axis). In a case in which the classification is carried out in a device with restricted capabilities, these features must be as easily computed as possible. To be able to carry out classification, a previous training phase is necessary, so the algorithm learns the optimal way to assign the corresponding class. This stage relies on a training dataset (see Sect. 4.1) of instances created with sample data, which must reflect real operation conditions as much as possible.

Figure 2 shows an overview of our activity recognition system. Basically, it is prepared to handle two classification stages. The first one (marked as *Stage 1*) focuses on calculating the relative position of the mobile device with respect to the user's body (e.g., the output of this stage should be a statement like 'the smartphone is in the trouser's front pocket'). In the second one (marked as *Stage 2*), the activity is recognized relying on sensors' data for a given position (e.g., 'the user is running'—additional information on speed, etc. may be added). Whether *Stage 1* may be present or not, in the following sections we are going to study the benefits and costs of including this stage. In case *Stage 1* is not present, the activity is estimated using classifiers trained with all the data gathered in different positions. As the reader will notice, in *Stage 2,* there are many feasible combinations in terms of sensor signals, features, classifiers and sliding window duration; each combination will offer different estimation accuracies (also dependant on whether the first classification stage is active or not) at different computational and time delay costs. For example, the activity estimation system may be working: (i) *with device's position estimation stage*, *on all the feature set from every available inertial and orientation sensor*, *processing information from all the axes of each sensor*, *using sliding windows with overlap*, *and a Decision Tree classifier*; or (ii) *without device's position estimation stage*, *on mean and variance features of acceleration vector*, *using non-overlapping sliding windows*, *using Naïve Bayes classifier*. Presumably, the first option will offer better accuracy for activity estimation as well as shorter delay detection, but it will demand more resources in the mobile device and more computation time than the second one. The activity recognition system's configuration choice will depend on the characteristics of the hosting mobile device and on the application requirements.

The classifiers used in this work are Naïve Bayes, Decision Table and Decision Tree, which have been chosen due to their simplicity and good performance in terms of computational delay and weight.

The Naïve Bayes algorithm follows the Naïve Bayes simplified formula and uses normal distribution for numeric attributes [38]:

$$p(C=c/X=x) = \frac{p(C=c)\cdot p(X=x/C=c)}{p(X=x)} \cong p(C=c)\cdot \prod_i p(X_i=x_i/C=c)$$

where C is the classification class, and X is the set of features taken into consideration for the classification.

Decision Tables [39] associate conditions of the values of the attributes considered with a classification result. The algorithm used to search the best group of attributes is best first, which begins with an empty set of attributes and adds features until the best set is found.

A Decision Tree [40] (J48, an implementation of the C4.5 classifier) is a tree-like set of nodes. In each node, a condition related to the value of an attribute is checked. In this way, depending on whether the condition is fulfilled, the condition of the following node is checked until a leaf

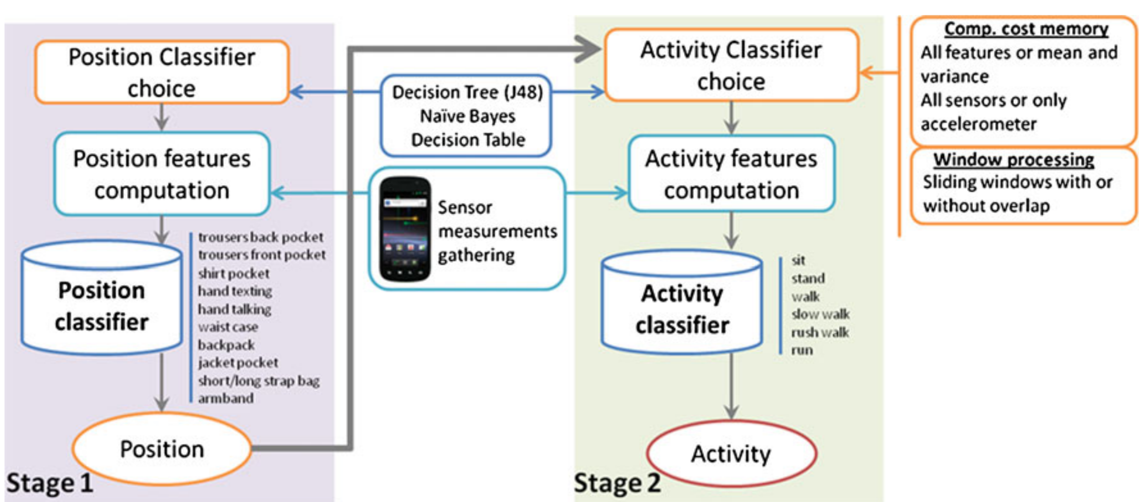

Fig. 2 Activity classification system overview: *Stage 1* estimates the on-body position (optional); *Stage 2* performs activity classification

that contains the classification result is found (the options to prune the tree with a 0.25 confidence threshold, as well as a minimum number of leaves per node of 2 are chosen).

Naïve Bayes (the simplest classifier) has been considered as a baseline to compare the rest with it. A priori, Decision Tables and Decision Trees demand more memory resources than Naïve Bayes. Decision Trees are expected to classify better than Decision Tables since they are not symmetrical: a given condition based on one attribute value can be crucial for the classification carried out along one branch, whereas on another branch, it may not appear because it does not add valuable information.

In all cases, the training phase is to be carried out outside the device, and the implementation of the models resulting from it will be the ones that are loaded for online recognition. As it is shown in Sect. 5.3, the size of the classifier that estimates the position may be large (>5 MB), and the device may lack memory to load it. Additionally, if the activity estimation system includes *Stage 1* for on-body device's position estimation, it will be necessary to handle different classifiers for each position, so strategies to optimize the memory use will have to be explored.

For our experiments, all the embedded sensors in the smartphone are considered (accelerometer, gyroscope, gravity, linear acceleration, magnetometer and orientation). Anyway, relying on a single accelerometer makes the classification process less demanding in terms of computational load.

We next comment on the different design variables to consider in each classification stages.

4.3 Stage 1: determining the mobile phone position relative to the user's body

Obviously, signals gathered when performing a specific activity depend on where the mobile device is being carried. In this way, it is expected that a classifier working on data from a mobile device in a known position will be able to better differentiate the activities.

To estimate the position of the mobile device, we have relied on instantaneous raw measurements (from the dataset mentioned in Sect. 4.1), instead of using sliding windows considering time slots. This is because it is desirable that on-body position detection is carried out with the shortest possible delay.

Light and proximity sensors complement the information coming from inertial systems when inferring the device on-body position. We use light sensors to determine whether the device is into a pocket, bag (darkness) or outside (light) (in practice, calibrating a threshold has been enough to differentiate the light-darkness states). Nevertheless, the individual could be in a dark room (e.g., the cinema), so additional information is necessary to make that decision. The proximity sensor fires whenever an obstacle is closer than 3 cm. This is useful when the mobile is placed inside a bag, case or a pocket, and also when the user is talking on the phone.

The magnetometer helps determining the orientation, which is a significant feature to distinguish among sets of on-body device positions: vertical (pocket, armband), horizontal (waist case) or random (bag, backpack). The projection of the gravity vector onto the coordinate axis also measures the orientation of the device, so this information is used for the same purpose.

All in all, two sets of signals have been considered for the analysis. Firstly, a set composed of projection of gravity onto x, y, z axes, light and proximity. The second set of features gathers the best ones among all: the most correlated ones with the class attribute and the most uncorrelated ones among them. Thus, the next 10 sensor signals are processed: projection of the gravity onto the three axes (x, y, z), proximity, light, accelerometer (y and z axes) and magnetometer (y and z).

The classifier to estimate on-body device position is finally chosen among the three available considering the requirements of the application and the limitations of the device.

4.4 Stage 2: activity recognition and influence of applications and device constraints

Stage 2 will throw activity estimations as output. Below, we comment on four design choices that condition the performance (accuracy, delay and resource consumption) of the *Stage 2* in the activity recognition system: (i) signal feature selection, (ii) classifier selection, (iii) how sliding windows are used, (iv) effect of including *Stage 1*.

4.4.1 Signal feature selection

Let us recall again the list of mobile sensors that may be used to classify activity: accelerometer, gravity, linear acceleration, gyroscope, magnetometer and orientation. In addition to the values gathered on each axis, the magnitude of the vector of measurements for each sensor is computed every time a new measurement is received.

In the training phase, a set of features is computed on these data. The features considered have been divided into time- and frequency-domain features (having the latter higher computational cost since they imply the computation of a Fourier transform). The energy of the signal has been computed as well. The complete list of features (Table 4) has been created taking into consideration previous works (Sect. 2.2). The method that looks for the attributes that are highly correlated with the class attribute, and are as uncorrelated as possible among them, is used to choose the significant features.

Table 4 Time and frequency-domain features

Time-domain features	Frequency-domain features
Mean	Power spectrum centroid
Variance	FFT energy
Zero crossing rate	Frequency-domain entropy
75 percentile	
Interquartile	
Correlation between acceleration and gyroscope, acceleration and magnetometer, linear acceleration and gyroscope, linear acceleration and magnetometer	
Signal energy	

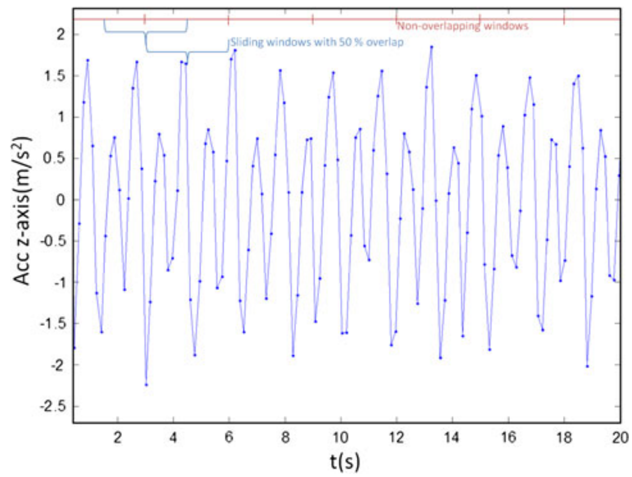

Fig. 3 Values of the *z axis* of the accelerometer during a slow walking test

4.4.2 *Classifier selection to reduce computational cost and memory fingerprint*

To reduce computational cost and memory load, it is possible to choose classifiers trained with the best mean and variance features, both when considering all the sensors or the accelerometer only, instead of computing all the features.

We have also considered the possibility of just using acceleration features. This single-sensor approach reduces memory needs and computational cost, and in addition, increases the number of devices (not only smartphones, but also nodes of wireless sensor networks) in which the system may be deployed.

4.4.3 *Using sliding windows to compute the features*

In our case, the features are computed over a temporal window (3 s long). The length of the window has been chosen after analyzing the walking periods at different paces for the different users in our dataset. When walking slowly, one step may take 1.15 s, so 3 s is enough to get several steps into the processing window regardless of the walking pace. Figure 3 shows a signal gathered from the z-axis of the accelerometer during a slow walking test and the samples in the sliding windows (with and without overlap). Short-duration activities could be overlooked using this window length, but detecting transitions is not a design requirement for our system: such short movements are not significant for most life-logging applications, which usually rely on slot-aggregated information (see Sect. 6).

Computing features using sliding windows with or without overlap is a design decision that has an impact on the estimation delay and in the computational cost. In the next section, both approaches have been explored, in order to get accuracy results when using both possibilities.

4.4.4 *Including stage 1 (on-body position estimation stage)*

If *Stage 1* is active in the recognition system, then *Stage 2* will work on classifiers trained using data gathered for the detected on-body device position. If not, the classifier for *Stage 2* will be chosen among the ones trained for all the sample data together (regarding the limitations of the device and the application). Obviously, these general classifiers require more memory space, since they try to estimate the same activity by using the samples from all positions, and the movements sensed in different positions present significant differences.

Section 5 considers each of these four aspects and analyzes the practical effect of each choice.

5 Evaluation of the classification process

The method *leave-one-subject-out* has been used to get all the results, because it separates the data from the user whose results are computed from the training set. The *leave-one-subject-out* method implies that for each individual, two files have to be created: the first one contains the data corresponding to all the individuals of the database, except for the one that is being considered and is used to train the classifier, and the second one contains the data corresponding to that individual and is used to test the classifier. All the results obtained for each individual are then averaged to get the final result.

Next, results thrown by the different classification algorithms to estimate activity as well as position are covered. In practice, WEKA tool [41] has been used to train the classifiers and get the results.

5.1 Activity recognition and influence of applications and device constraints

The results presented next consider all the possibilities mentioned in Sect. 4.4, in terms of combinations of sensor signals, classifiers, feature selection and windows with or without overlap. In order to enhance clarity, each subsection focuses on one of these single aspects and analyzes it.

To better compare the results, some tables have been included: Tables 5 and 6 show the accuracies of all classifiers for all the specific cases that will be considered in the subsections below. The cases in which all the sensors are used to get the results are distinguished from those only based on the acceleration data. Tables 7 and 8 contain the average number of leaves and the tree sizes (number of nodes) of the Decision Tree classifiers and the number of rules of the Decision Table, for the results computed considering all sensors and the accelerometer, respectively. Obviously, these aspects are related to the practical size of the codified algorithm.

The comparison of the results of different algorithms as well as the comparison of the results of classifiers with different characteristics are done using the Mann–Whitney *U* test and the Wilcoxon signed rank sum test [42]. Both tests are non-parametric and compute the *p* values to test a null hypothesis (i.e., the probability of obtaining a statistic at least as extreme as the one observed in the results, under the assumption that the null hypothesis is true). The Wilcoxon signed rank sum test is applied for paired data and tests the null hypothesis that the median of the difference is

Table 5 Average results taking into account all the available sensors

All sensors	Sliding windows without overlap			Sliding windows with overlap		
	DT	NB	J48	DT	NB	J48
Best set composed of all features						
Stage 2 classification	78.36	67.85	93.14	77.83	70.23	92.29
Stage 1 + Stage 2 classification	89	84.17	96.92	88.63	83	97.02
Best set composed of mean and variance features						
Stage 2 classification	75.76	62.14	86.17	78.12	62.81	87.51
Stage 1 + Stage 2 classification	88.15	78.20	96.26	88.96	78.41	96.68

Table 6 Average results using only information from the accelerometer

Accelerometer	Sliding windows without overlap			Sliding windows with overlap		
	DT	NB	J48	DT	NB	J48
Best set composed of all features						
Stage 2 classification	75.87	66.97	88.16	76.01	66.49	88.60
Stage 1 + Stage 2 classification	86.20	81.0	94.82	86.35	78.56	95.43
Best set composed of mean and variance features						
Stage 2 classification	75.86	61.43	78.48	75.91	58.36	80.23
Stage 1 + Stage 2 classification	84.25	73.14	91.67	85.65	72.54	91.44

Table 7 Average number of rules of the Decision Table and number of leaves and size (number of nodes) of the Decision Tree classifiers considering all the sensors

All sensors	Sliding windows without overlap			Sliding windows with overlap		
	DT	J48		DT	J48	
	No. of rules	Size	No. of leaves	No. of rules	Size	No. of leaves
Best set composed of all features						
Stage 2 classification	1507.1	2030.1	1015.6	797.44	4011.4	2006.2
Stage 1 + Stage 2 classification	154.15	93.19	47.08	259.58	146.33	73.65
Best set composed of mean and variance features						
Stage 2 classification	285.19	1183.6	592.31	1982.6	2060.6	1030.8
Stage 1 + Stage 2 classification	169.48	90.09	45.53	287.82	133.87	67.42

Table 8 Average number of rules of the Decision Table and number of leaves and size of the Decision Tree classifiers only considering the accelerometer

Accelerometer	Sliding windows without overlap			Sliding windows with overlap		
	DT	J48		DT	J48	
	No. of rules	Size	No. of leaves	No. of rules	Size	No. of leaves
Best set composed of all features						
Stage 2 classification	266.75	1949	975	361.81	3698.8	1849.9
Stage 1 + Stage 2 classification	97.95	124.53	62.75	197.78	211.29	106.13
Best set composed of mean and variance features						
Stage 2 classification	212.63	477.63	239.31	323.88	1117.6	559.31
Stage 1 + Stage 2 classification	96.75	101.88	51.42	186.50	149.91	75.43

zero. It also relies on the assumption that the distribution of the differences is symmetric, which can be applied to our case. To compute the p values, the difference in the values is sorted and the sum of the ranks of the positive and negative differences is computed. Then, the minimum of those values is chosen to compute the p value using a normal distribution (since the number of results $N = 16$ is high enough to use the normal approximation of the distribution). The Mann–Whitney U test sorts the results of both distributions and computes for each value of each distribution the number of values of the other distribution that are smaller than it. Then, for each distribution, these numbers are added, and the minimum is chosen to compute the p value using again the normal distribution approximation. If the p value is under a specific value (usually 5 %), the null hypothesis is considered to be false; otherwise, more data are needed to conclude that the null hypothesis is true.

With respect to the notation and the expressions used in the tables, DT stands for Decision Table, J48 is the algorithm applied for the Decision Tree and NB stands for Naïve Bayes. ‘All sensors’ means that the features computed using the signals from the accelerometer, gravity, linear acceleration, magnetometer, orientation and gyroscope are considered to train the classifiers. On the other hand, ‘ACC’ includes the results achieved when the features computed using only the accelerometer signals are considered. ‘Best set composed of all features’ stresses that all the time- and frequency-domain features included in Table 4 are fed to the method that chooses the best attributes to carry out the classification. In addition, ‘best set composed of mean and variance features’ means that only mean and variance features are considered in the method that chooses the best ones to carry out the classification.

By ‘Stage 2 classification,’ we mean that the classification does not include the stage to estimate the position of the mobile, that is, the classifiers have been trained with data gathered from the different positions together (Fig. 2). Moreover, ‘Stage 1 + Stage 2 classification’ shows that a first stage has detected the position of the smartphone, and the classifier is trained using information gathered on that specific position. The results included in the tables are the ones calculated averaging the ones obtained in all positions.

5.1.1 Sensors and feature analysis

Let us first analyze which sensor signals and features are the ones most commonly used to perform activity recognition in our system. Before training the algorithm, there is a processing stage to choose the features that are the most correlated with the class attribute and the most uncorrelated among them (this is implemented in the CfsSubsetEval method in WEKA [41]). One example of attributes in the case in which de position of the mobile has been detected to be the ‘trousers front pocket,’ and the accelerometer that is the only sensor considered is variance, percentile 75, interquartile, energy and frequency entropy of the y-axis signal, variance, percentile 75, interquartile and power spectrum centroid of the magnitude signal, and zero crossing rate and energy of the fft of the z axis signal.

Table 9 contains the percentage of acceleration-derived features chosen (% Acc) (i.e., they are computed using the measurements from acceleration, gravity or linear acceleration). The percentage of frequency-domain features that are considered is also included (% Freq). Regarding the nomenclature, NFeat stands for the number of features used in that case.

With respect to the percentage of features that rely on the information of the accelerometer (including gravity and linear acceleration sensors) represent the 66 % in the classifiers when *Stage 1* and *Stage 2* of the system are performed, and the 75 % in the classifiers that only consider the *Stage 2*. This means that the accelerometer is the sensor that more information contains to differentiate the activity.

Finally, the percentage of frequency-domain features used to obtain the results is around 30 % in the highest

Table 9 Number of features and percentage of accelerometer and frequency-domain features used to train the classifiers

Sliding windows without overlap	All sensors			Acc	
	% Freq	% Acc	NFeat	% Freq	NFeat
Best set composed of all features					
Stage 2 classification	31.8	72.7	22	8.3	12
Stage 1 + Stage 2 classification	25.1	66.7	30	15.0	13
Best set composed of mean and variance features					
Stage 2 classification	–	75	12	–	4
Stage 1 + Stage 2 classification	–	63.1	13	–	5

case, so this means that time-domain attributes provide more information than frequency ones to classify the activity.

5.1.2 *Classifier performance*

In order to compare the results from the different classification methods trained with the same type of parameters, the Wilcoxon signed rank sum test has been used [42]. The data of the different algorithms are paired, since the results for each individual are obtained using the same test and training datasets for the three classifiers considered (Decision Tree, Decision Table and Naïve Bayes). The classifiers are compared in groups of two, that is, Naïve Bayes vs. Decision Tree, Naïve Bayes vs. Decision Table and Decision Table vs. Decision Tree. The two-sided p values obtained are all under 1 %, so there is strong evidence that in all the cases, the Tree classifier outperforms the Decision Table and the latter outperforms the Naïve Bayes results, so the average results of the classification are not just a coincidence for those specific datasets. The Decision Tree classifier throws a best accuracy value, around 94 % (considering all the sensors and the best set composed of all the features) and in the worst case around 78 % (activity classified performing only the *Stage 2*, considering only the accelerometer and the best set that contains mean and variance features). The Decision Table classifier accuracy varies between 89 % and 75 %, and the Naïve Bayes classifier between 84 and 60 % in the same cases.

Regarding the characteristics of the confusion matrices for the different classifiers:

- Naïve Bayes algorithm misclassifies *Standing* instances as *Sitting* ones or vice versa a percentage of times that varies with the characteristics of the classifier but that can be as high as 90 %, for instance in positions such as the shirt pocket in which there is no difference in the position of the mobile when *Standing* and *Sitting*.
- Walking at different paces activities are naturally most probably confused with the corresponding lower and higher speed. The percentage of the cross-classification varies depending on the characteristics of the classifier. Significant values will be analyzed in the following subsections.
- The *Running* activity is detected with over 85 % accuracy in almost any case. The only exception is when using a Decision Table and carrying the mobile phone in the trousers front pocket.

5.1.3 *Influence in the classifier accuracy of the device's position with respect to the user's body*

Following, there is an analysis of the effect of having a priori information on where the mobile device is placed with respect to the user's body. In order to compare the results, the Mann–Whitney U test is used [42]. In this case, the results are not paired, and the mean accuracy value of all the positions for each person is compared to the value achieved using the datasets that do not differentiate the position of the mobile. The same type of classifiers is compared at each time (e.g., the results achieved using Decision Tables with the set composed of the best features of all sensors and *Stage 2* only classification vs. *Stage1* plus *Stage 2* classification). The results of the two-sided p values computed are under 1 % in every case, so there is strong evidence that the classifiers that take into consideration the position outperform the ones that classify regardless of it.

Regarding the confusion matrices, it can be observed that:

- The trousers front pocket position is the position with lower accuracy when classifying the activities.
- In the case of the Decision Tree algorithm, the *Normal Walking* and *Rush Walking* activities are more cross-classified in the trousers front pocket than in other positions.
- The classification of *Standing* and *Sitting* activities is improved significantly when considering the position in which the mobile is being carried, and the features computed from the information received from all the

sensors. In the case in which only the accelerometer is used, the classification of those activities is also improved, although the accuracy in general is reduced.

Once it is proven that knowing the position in which the mobile is placed enhances the results, Sect. 5.2 explores to which extent these positions can be distinguished.

5.1.4 Computational cost and memory-load analysis

As it has been covered in Sect. 4, in order to reduce the computational cost of the algorithms and the memory constraints, two possibilities have been taken into account.

5.1.4.1 Using mean and variance as features for choice Instead of using the set composed of best features chosen among all, the search is only carried out among the *mean* and *variance* features (which are computationally low-cost ones).

The Mann–Whitney U test is used [42] in the same way as in Sect. 5.1.3. The results achieved choosing the set composed of best features among all the ones available and the set composed of best mean and variance features are compared for the same type of classifiers. The two-sided p values are in many cases over 5 %, so there is no evidence that in those cases, one classifier outperforms the other, and more experiments will be necessary to determine the specific behavior. These cases are the ones in which the position and all the sensors are considered for the Naïve Bayes and Decision Tree classifiers and the results of the Decision Table in any case.

With respect to the confusion matrices, it must be remarked that:

- The classification of the *Running* activity remains almost constant in most of the classification algorithms when only the mean and variance features are considered.
- In the case of Naïve Bayes algorithm, the accuracy of the classification of all the activities (except for *Running*) is reduced.
- In the case of Decision Tree, the accuracy of the classification is mainly reduced in the *Standing* and *Sitting* cases when all the sensors are considered.

5.1.4.2 Using features computed from the accelerometer signals only There are still many devices in the market which are only equipped with a single accelerometer, which otherwise has come to be the richest information source regarding activity (Sect. 5.1.1). In order for our system to be scalable and run in as many devices as possible, results using only the accelerometer have also been computed.

As the results are not paired, the Mann–Whitney test is again used [42], and the measurements from all the sensors are compared to the accelerometer ones. It can be observed when computing the p values that in many cases, there is significant evidence that using all the sensors outperforms the results achieved using only the accelerometer; some exceptions are the results corresponding to Decision Table and Naïve Bayes except when mean and variance features are considered, in which more experiments should be carried out to decide whether the results are really similar.

Analyzing the confusion matrices, it must be noted that:

- The lost of precision when considering the accelerometer sensor is due to the low accuracy when cross-classifying *Sitting* and *Standing* activities when all the features are considered in the case of the Decision Tree algorithms. The same phenomenon is shown in the Decision Tables when the *Stage 1* and *Stage 2* are considered to classify.

5.1.5 Using sliding windows to compute the results

As stated in Sect. 4.4, features have been computed using a 3-s duration window with and without overlap.

In order to compare the results, the Mann–Whitney test is again used [42]. The results achieved using sliding windows with overlap to process the measurements are compared to the ones without overlap. All the two-sided p values computed are over 3 %, so there is no evidence that one classifier outperforms the other. It can be concluded that there is no significant evidence that the sliding windows with and without overlap produce different results.

Regarding the confusion matrices, it can be observed that:

- Decision Tree: the accuracies remain almost constant but in the case in which only the *Stage 2* is considered, more variations are noticed.
- Naïve Bayes: variations do not follow a specific pattern; the accuracies remain constant in most of the cases.

5.2 Determining the mobile phone position relative to the user's body

Section 5.1.3 shows how activity recognition can be enhanced by a priori knowing where the mobile device is being carried. Here, we explore to which extent this relative position can be estimated.

Results in Table 10 show that much better accuracy can be obtained taking into consideration the set of features composed of the selected best features (over 90 %

Table 10 Average results for position classification

	Decision Table	Naïve Bayes	Decision Tree
Best set of all features	89.76	54.45	92.94
Selected features	66.61	50.67	66.50

Table 11 Average number of tree leaves, tree size (number of nodes) and number of rules for position classification

	Decision Table	Decision Tree	
	No. of rules	Size	No. of leaves
Best set of all features	305530	31818	15909
Selected features	2881.3	1905.6	953.31

accuracy with the Decision Tree classifier), but regarding the number of rules and the size of the tree (Table 11) and the sizes of the binary files in which the classifiers are converted (Table 13), this option implies dealing with much bigger trees than when some selected features are chosen.

To conclude, it is important to remark that this classification of the position relies on other sensors apart from the accelerometer, so a new algorithm must be developed to distinguish the position in the case when only this sensor is available. At this moment, this kind of devices must classify the activity regardless of the position with the loss of accuracy shown in the corresponding tables.

5.3 Integration of classifiers in a smartphone

To integrate the classifiers in a smartphone (on Android OS), they have been converted into Binary Tree, Decision Table and NaiveBayes java objects that are stored in a binary file and loaded whenever the classifier is necessary in the application. The implementation of the classifiers as objects allows the java Garbage Collector to delete the created object if it has not been used for a while, releasing the corresponding memory from the mobile phone. All these objects implement an interface that contains a method to inform the application which features must be computed and a second method to carry out the classification.

Tables 12 and 13 show the sizes of the binary files that contain the implementation of the different classifiers. As expected, the smallest files are the ones corresponding to Naïve Bayes. When considering the classifiers that estimate the position in which the mobile is being carried, the size of the Decision Table is too big to be loaded into the mobile (Table 13). On the other hand, the sizes of the classifiers that estimate the activity are small (Table 12). In addition, the sizes of the classifiers that consider only *Stage 2* are bigger than the ones that estimate first the position (*Stage 1* + *Stage 2*). Finally, when considering the set only composed of mean and variance features, the reduction is not significant.

Traceview Android tool is used[1] to measure the necessary average times to run feature extraction and subsequent classification. This tool allows storing the start/stop execution times of each thread and method together with a summary of the time spent in each inner method call. The measurements are repeated 50 times to reduce the influence of outliers; results are then averaged. We are aware that these results are device-dependent: as it is stated in the tool documentation, the information provided is not valid as absolute timings, but useful when considering other results provided by the same tool. In this way, we use the tool to compare which are the methods that take longer to be computed and under which circumstances it could be expected that the classifier carries out the classification faster.

Tables 14 and 15 show the average times obtained using the Traceview tool. As expected, computing instantaneous samples of the sensors to estimate the on-body device position is less demanding than estimating activity. The classification stage takes less time than computing the features, which is coherent, since the classification process is composed of comparisons and multiplications in the case of Naïve Bayes, whereas when computing the features, more computations are involved, even in the simplest ones such as the mean and the variance. It can also be noted that the classification stage of the Tree classifier is faster than that of the Decision Table, but the former relies on more features to accomplish the classification, which takes longer than in the latter case. The time reduction is significant when only mean and variance features are considered, due to the fact that their computation load is lower. The computation load when considering only the attributes using information from the accelerometer is lower than when using information from all the sensors.

5.4 Design rules

Following, some guidelines to design a mobile activity recognition system are presented (regarding all the information commented throughout this section):

- The Decision Tree is the classifier that provides better results in terms of accuracy, considering both types *Stage 2* and *Stage 1* + *Stage 2* classifiers. The set of features is composed of the best time and frequency-domain features computed using the information gathered from all the sensors.

[1] From Android developers: http://developer.android.com/guide/developing/debugging/debugging-tracing.html.

Table 12 Sizes of the binary files that contain the implementation of the activity classifiers as java objects

Sliding windows without overlap (kB)	All sensors			Accelerometer		
	DT	NB	J48	DT	NB	J48
Best set composed of all features						
Stage 2 classification	65.7	1.9	57.7	8.4	1.2	55.4
Stage 1 + Stage 2 classification	5.4	2.4	3.0	3.7	1.3	3.8
Best set composed of mean and variance features						
Stage 2 classification	10.0	1.2	33.9	7.0	0.7	14.2
Stage 1 + Stage 2 classification	5.1	1.2	2.9	3.6	0.7	3.3

Table 13 Sizes of the binary files that contain the implementation of the classifiers that estimate the position of the mobile as java objects

Decision Table	Naïve Bayes	Trees
16.5 MB	1.46 kB	873.6 kB

Table 14 Time measured when computing the features and carrying out activity classification

All axes		Sliding windows without overlap					
		All sensors			Accelerometer		
		DT	NB	J48	DT	NB	J48
Best set composed of all features							
Stage 2 classification	Time computing features (ms)	19.94	2680.0	2694.8	19.45	827.26	845.93
	Time classify (ms)	10.62	7.24	0.14	8.83	3.92	0.084
Stage 1 + Stage 2 classification	Time computing features (ms)	139.95	1905.16	1908.7	254.88	1272.75	1028.47
	Time classify (ms)	4.59	30.72	0.086	2.58	4.23	0.064
Best set composed of mean and variance features							
Stage 2 classification	Time computing features (ms)	16.059	108.70	107.51	16.54	33.99	34.19
	Time classify (ms)	9.42	3.98	0.071	10.44	1.33	0.090
Stage 1 + Stage 2 classification	Time computing features (ms)	17.78	104.24	87.82	16.89	41.54	41.51
	Time classify (ms)	6.068	4.17	0.056	4.18	1.67	0.065

Table 15 Time related to the computation and classification stages when classifying the position of the mobile

Time	Naïve Bayes	Trees
Time computing features (μs)	82.96	84.35
Time classifying (ms)	5.12	0.159

- The classifiers that compute *Stage 1* and *Stage 2* (Fig. 2) have better accuracy, but it has an extra delay because it is necessary first to estimate the on-body device's position and then to load the classifier corresponding to that position. Moreover, the delay due to the load of the classifier that estimates the position must be considered, although it is only relevant when the application is launched.
- When there are battery restrictions, a solution based only in the accelerometer is suitable, since the computational load is reduced. Additionally, as the number of sensors the application has to be subscribed to is also smaller, the power consumption diminishes.
- In case that the CPU usage of the device is a limiting feature, the solutions (i) in which only the signals coming from the accelerometer or (ii) based on mean and variance features should be chosen to reduce the computational load. The case that considers only the accelerometer saves battery power since the application only has to be subscribed to its measurements.
- Using sliding windows with or without overlap results in similar accuracy and computational complexity, so

the delay constraints must be analyzed to choose one case or the other.
- Memory constraints are not significant in any case, because the sizes of the files are not too restrictive.

6 Activity recognition in practice: design and demonstration of an activity monitor

In order to apply the conclusions of our previous analysis, we have developed the activity recognition architecture described in Sect. 4 to support the functionalities of an Activity Monitor: a 'persuasive' mobile application capable of tracking and evaluating the user's physical activity all day long, in order to (i) keep him aware of his habits and (ii) help him to modify unhealthy inactive patterns. Next we describe the design details for the application, together with the results of some experiments in real conditions that demonstrate how the activity recognition system finally works.

6.1 Translating activity into energy expenditure

To build our Activity Monitor, it is key to determine how to translate activity estimates into relative energy expenditure; basically, this translation will make possible to compare the instantaneous and accumulated activity levels with personalized thresholds, to evaluate the user performance. Thus, following, there is some background work that supports our choice of design to deal with activity-energy conversion.

The relationship between energy expenditure and physical activity has been widely explored in health studies, existing several indexes to quantify it. The total amount of energy a person consumes is known as TEE (Total Energy Expenditure). It is composed of (i) basal metabolic rate (BMR), the minimal amount of energy a body requires when lying in physiological and mental rest, (ii) the thermic effect of food (usually considered to be 10 % of BMR [43, 44]) and (iii) the physical activity–related energy expenditure (AEE) that includes energy consumption due to muscular activity and increased cardio respiratory function. AEE represents an absolute value that is influenced by people particularities (e.g., body weight) as well as by the activity performed, so this index cannot be used to compare activities intensity or duration among different individuals. Therefore, there are alternative indexes that make possible to compare activity levels among people with different physical characteristics: for example, MET–metabolic equivalent, PAR–physical activity ratio and PAL–physical activity level. PAR and PAL are the most used ones; they are complementary: if PAR is defined as 'the energy cost of an activity per unit of time (usually a minute or an hour) expressed as a multiple of BMR' and 'it is calculated as energy spent in an activity/BMR for the selected time unit' [45], then PAL is the time-weighted average of the PAR associated with the set of activities performed during 24 h [43].

The application presented in this work employs PAR and PAL indexes to quantify activity levels, as they only depend on the performed activity and not on the user characteristics. These indexes have been previously used in a variety of works for activity monitoring (e.g., [43–46]). It is important to remark that the research community continuously updates the tables mapping activities and PAR as more accurate methods for measuring energy expenditure are defined. Maybe the most notable effort in this sense is the 'energy costs of activities' tables of [43] (annex 5) that have been later updated or extended in [46].

6.2 Application architecture and activity recognition design

Let us bottom-up explore the modular architecture defined for the Activity Monitor (Fig. 4). The application needs to gather raw data from mobile sensors (*Acquisition Module*) in order to calculate the features needed by the classifier, which is in charge of inferring the activities the user is performing (*Classification Module*). The detected activity is then quantified into its corresponding PAR (according to "energy costs of activities" tables in [43], annex 5), taking also into account the activity duration (this is done in the *Quantification Module*). The accumulated activity level is constantly evaluated and compared with the user's personal activity level objective: a curve representing the user's desired activity level during the current day (*Evaluation Module*). Different types of notifications will be triggered if the actual accumulated activity level is below the expected objective; in this case, the *Notification Module* will generate a suitable type of feedback (logs or alarms, depending on the user's context).

With respect to the activity recognition system, it is important to take into account that (i) the application needs to be running continuously, so its memory fingerprint should be as small as possible; (ii) high accuracy is desired but not critical, as classification results are integrated over time; (iii) transitions between physical activities can be delayed. Taking into account results in Sect. 5 and the previous requirements, the Decision Tree classifier that does not take into consideration the on-body position, and works on the best set of time and frequency-domain features computed from all the sensors, has been selected for implementation, since it is expected to provide the most accurate classification results.

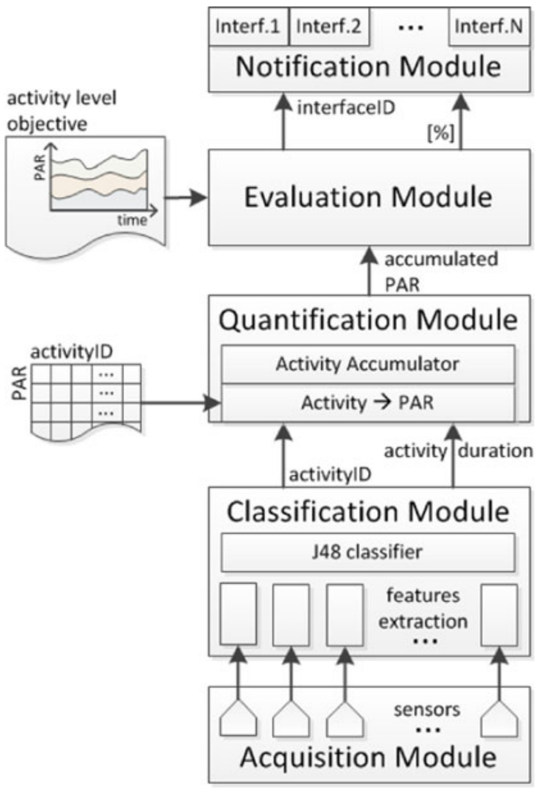

Fig. 4 Activity Monitor application modular design

6.3 Monitoring activity in a office scenario: demonstration on a real use case

We have implemented the Activity Monitor to run in Google Nexus S smartphone over 2.3.1 Android, in order to validate the activity recognition stage in a real use case: sedentariness level evaluation on white-collar job environments.

The Activity Monitor has been initially configured with different activity level curves $a_l(t)$, formally defining the required activity level during a certain period of time T (to be also initially configured: a complete day, working day hours, etc.), that is, a desired PAR curve. Although more complex curves could be used, in this version, we equally distribute the desired daily activity during the whole period of time initially configured.

Once the application is launched, it starts calculating the user's activity level $r(t)$ (taking into account the PAR associated with the activities, the user is performing). Two kinds of evaluations are performed periodically (every 3 s): (i) $m_l(t)$, the ratio between the real accumulated activity level $\left(R(t) = \int_o^t r(\tau)d\tau\right)$ and the desired accumulated activity level $\left(A(t) = \int_o^t a(\tau)d\tau\right)$ and (ii) $m_l^T(t)$, the ratio between the real accumulated activity level [$R(t)$] and the total desired activity level [$A(T)$]. These measurements will be used to inform the user about his/her activity level performance. Basically, the application log will be updated continuously, for the user to be able to check it, and a sound/vibrating alarm will be generated three times a day (lunchtime, mid-afternoon, and early evening), if the performance $m_l(t)$ decreases below the 25 % of the desired accumulated activity level for that moment.

To validate the application concept and the activity recognition system, two different users have been carrying the mobile device during a standard working day and have been requested to annotate their activity (the application was facilitating so for test purposes). No restrictions regarding the mobile device on-body position have been imposed to the users.

Figure 5 shows activity performance obtained from the Activity Monitor's logs for User 1 (left) and User 2 (right), depicting PAR values over the logging time.

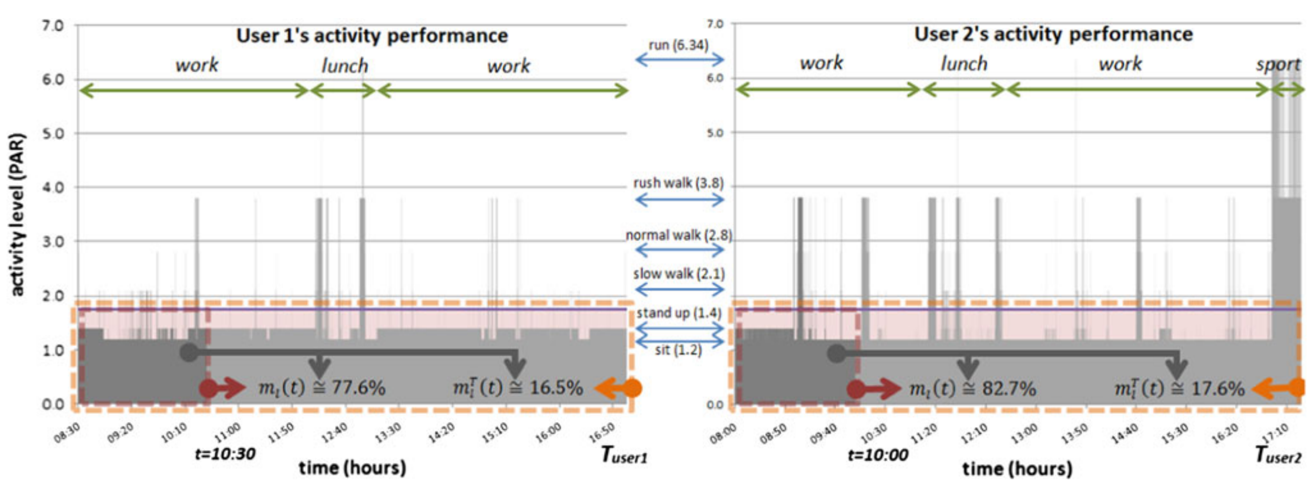

Fig. 5 *User 1's* (*left*) and *User 2*'s (*right*) activity performance in PAR during their working day

From the curves, User 1 seems to stay most of the time at his desk, moving just when strictly needed. User 2 moves oftener: the pattern includes walks for some minutes each hour. Additionally, for User 2, half an hour of jogging is also included. This evaluation coincides with the activity patterns annotated by the users.

If we compare both performances with the same reference—the desired activity level curve (e.g., 1.76 PAR; horizontal line)—it can be noticed that, at time *t* (i.e., 2 h after activity record started), User 2's performance is much higher than User 1's. Analyzing the percentages related to the performance achieved at *t* with respect to the expected activity level for that time, they show that User 2 accomplished 82.7 % of the desired activity level for until *t*, while User 1 just achieved 77.6 %. Individually, this is the most practical feedback that the user can have to decide activating him. In addition, when comparing the percentage of activity level at to the expected level at the end of the day, the percentages are significantly different: 16.5 and 17.6 %, respectively.

Making the users aware of their real activity habits and providing adaptive advice may be very useful both for wellness and pathology treatment, as it was stated in Sect. 2. The activity recognition system performs as expected from Sect. 6 analysis and provides accurate enough estimates for the application purposes. It is important to note that we have not aimed at building and validating a complete operative application here, but the core logic, which include activity recognition and relative energy expenditure computation. The block built is the basis for different targeted applications, which for sure can be enriched taking into account additional context parameters, both for performance evaluation and user feedback.

7 Conclusions

This paper gathers an analysis on how to build an activity classifying system that relies on smartphone's (inertial, proximity and light) sensors, ready to be built in the mobile device. As a whole, the system aims at offering a set of adaptive classification solutions that may offer a trade-off between accuracy and computational cost, without imposing any particular request on the user's side (the individual may place the device in any 'standard' body position, at his/her convenience). From our work, we conclude that it is possible to successfully use the device as physical activity sensor to log and evaluate the user's movements, in particular for those applications that integrate activity estimates over a period of time (e.g., our Activity Monitor). Applications needing real-time detection of movement transitions (being the time delay very critical) or needing to capture information from specific parts of the body (e.g., trauma rehabilitation) may have to rely on additional external wearable hardware.

The two-stage classification system includes a device position classification phase (*Stage 1*) to firstly determine where the mobile device is placed on the user's body. This stage significantly enhances the results achieved (Table 6) by the activity classifiers, but it is costly in terms of computational resources (Tables 12, 13, 14), so it is important to consider the application needs and device capabilities when considering if including it. For the activity recognition stage (*Stage 2*), some options have been explored to reduce computational load without aggressively compromising the accuracy. The first strategy is based on reducing the number of input sensors signals, by only using the accelerometer (which, in any case, is the most available sensor in many devices, not only phones). The second one controls the frequency of the activity feeds by the use of sliding windows with or without overlap for measurement. Windows' overlapping does not have a remarkable impact on accuracy. Finally, it is also possible to reduce the number of features to train the classifier or to simply use the mean and variance features instead of other time- and frequency-domain features. This strategy reduces the size of the classifiers models, having a small impact in the accuracy in the case of the Decision Table (Tables 5, 6).

With respect to scarce existing literature on stand-alone embedded mobile activity recognition, our work specifically focuses on three contributions: (i) an extensive analysis of the effect of estimating on-body positions for the mobile device (*Stage 1*). Additional on-body positions have been considered. (ii) An evaluation analysis method for classifiers' accuracy based on using *leaving-one subject-out*-strategy, which separates the set of data used for training the classifiers from the data used for testing, providing more reliable results than cross-classification method, (iii) a study on the practical implications of integrating classifiers in the mobile phone, measuring their memory needs and execution time. This aspect is relevant, because activity-logging applications are usually very demanding in terms of continuous computation, so their performance has to be optimized.

To demonstrate the practical use of our work, we have instantiated a recognition system for a mobile Activity Monitor, one of the many applications that may rely on mobile activity recognition. This application aims at 'logging, translating and evaluating the user's activity with respect to accumulated curves of relative energy expenditure.' Even if particularized for the specific case of use of monitoring and preventing sedentary habits in daily life, this monitor may be the initial core of other mobile applications related to health and wellness.

Directions for further work include enhancing the detection of the position in which the mobile is being carried, introducing sliding windows and time-domain features to estimate it. Profiling studies about the user's habits with respect to places to wear the mobile device could also be useful: for example, from our experiments, we noticed that the trouser's front pocket is the position where men most commonly choose to carry a mobile device, whereas women usually prefer bags. Regarding the different paces at which people perform the activities, a general pattern cannot be established since the speeds vary with the height as well as with the activity habits of the individual, so strategies to calibrate the system for a given user might also lead to better classification results.

Additionally, a methodology to facilitate the portability of the recognition system to other devices could be designed, by introducing a method that is able to adapt the classifiers to different frequency measurements and sensor ranges without losing accuracy significantly. It is also a pending task to explore the real power consumption of an Activity Monitor–type application, to evaluate whether it can be really compatible with making normal use of the device throughout the day (mainly due to battery).

Fusing activity recognition and location information may increase the accuracy of the activity recognition system, mainly to (i) deal with those idle moments in which the user is not carrying the device and (ii) subsequently infer high-level activity related to location (e.g., sleeping, driving a car, etc.).

Finally, there is still a lack of extensive studies on user experience, considering the practical problems that these mobile applications may suffer 'in the wild' and which may hamper adoption and adherence. We expect to contribute to this open issue in future works.

Acknowledgments Authors acknowledge their lab colleagues for the help provided by the persons who have collaborated on building the activities datasets, and Jesús Cano for his help developing the Activity Monitor. Henar Martín acknowledges the Spanish Ministry of Education for her grant. This work has been supported by the Spanish Ministry for Science and Innovation under grant TIN2011-28620-C02-02 and IPT2011-1052-390000 and the Government of Madrid under grant S2009/TIC-1485 (CONTEXTS).

References

1. Harun H (2011) Smartphone penetration in Asia set to boom. Nielsenwire. http://blog.nielsen.com/nielsenwire/global/smartphone-penetration-in-asia-set-to-boom/. Accessed 28 Sept 2011
2. Kellogg D (2011) 40 percent of U.S. mobile users own smartphones; 40 percent are android. Nielsenwire. http://blog.nielsen.com/nielsenwire/?p=28790. Accessed 28 Sept 2011
3. Lane N, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell A (2010) A survey of mobile phone sensing. IEEE Commun Mag 48(9):140–150. doi:10.1109/MCOM.2010.5560598
4. MediaCT GI (2011) The mobile movement study. http://www.thinkwithgoogle.com/insights/library/studies/the-mobile-movement/
5. Barkhuus L, Polichar VE (2011) Empowerment through seamfulness: smart phones in everyday life. Pers Ubiquitous Comput 15(6):629–639. doi:10.1007/s00779-010-0342-4
6. Tsai CC, Lee G, Raab F, Norman GJ, Sohn T, Griswold WG, Patrick K (2007) Usability and feasibility of PmEB: a mobile phone application for monitoring real time caloric balance. Mobile Netw Appl 12(2–3):173–184. doi:10.1007/s11036-007-0014-4
7. Bao L, Intille S (2004) Activity recognition from user-annotated acceleration data. In: Ferscha A, Mattern F (eds) Pervasive computing. Lecture notes in computer science, vol 3001. Springer Berlin/Heidelberg, pp 1–17. doi:10.1007/978-3-540-24646-6_1
8. Martin H, Bernardos AM, Tarrio P, Casar JR (2011) Enhancing activity recognition by fusing inertial and biometric information. In: Proceedings of the 14th international conference on information fusion (FUSION 2011), Chicago, IL, USA, 5–8 July 2011. IEEE, pp 1–8
9. Karaman S, Benois-Pineau J, Megret R, Pinquier J, Gaestel Y, Dartigues JF (2011) Activities of daily living indexing by hierarchical HMM for dementia diagnostics. In: Proceedings of the 9th international workshop on content-based multimedia indexing (CBMI 2011), Madrid, Spain, 13–15 June 2011. IEEE, pp 79–84
10. Philipose M, Fishkin KP, Perkowitz M, Patterson DJ, Fox D, Kautz H, Hahnel D (2004) Inferring activities from interactions with objects. IEEE Pervasive Comput 3(4):50–57
11. Blum M, Pentland A, Troster G (2006) InSense: interest-based life logging. IEEE Multimed 13(4):40–48
12. Puangpakisiri W, Yamasaki T, Aizawa K (2008) High level activity annotation of daily experiences by a combination of a wearable device and Wi-Fi based positioning system. In: Proceedings of the IEEE international conference on multimedia and expo (ICME 2008), Hannover, Germany, 23–26 June 2008. IEEE, pp 1421–1424
13. Matic A, Osmani V, Popleteev A, Mayora-Ibarra O (2011) Smart phone sensing to examine effects of social interactions and non-sedentary work time on mood changes-modeling and using context. In: Lecture notes in computer science, vol 6967. Springer Berlin/Heidelberg, pp 200–213. doi:10.1007/978-3-642-24279-3_21
14. Iglesias J, Bernardos A, Tarrío P, Casar JR, Martín H (2011) Design and validation of a light inference system to support embedded context reasoning. Pers Ubiquitous Comput 1–17. doi:10.1007/s00779-011-0447-4
15. Consolvo S, McDonald DW, Toscos T, Chen MY, Froehlich J, Harrison B, Klasnja P, LaMarca A, LeGrand L, Libby R, Smith I, Landay JA (2008) Activity sensing in the wild: a field trial of ubifit garden. In: Proceedings of the 26th annual SIGCHI conference on human factors in computing systems (CHI 2008), Florence, Italy, 5–10 April 2008. ACM, pp 1797–1806. doi:10.1145/1357054.1357335
16. Denning T, Andrew A, Chaudhri R, Hartung C, Lester J, Borriello G, Duncan G (2009) BALANCE: towards a usable pervasive wellness application with accurate activity inference. In: Proceedings of the 10th workshop on mobile computing systems and applications (HotMobile 2009), Santa Cruz, CA, USA, 23–24 Feb 2009. ACM, pp 1–6. doi:10.1145/1514411.1514416
17. Health Priorities (2003) The role of physical activity-public health physical activity background paper
18. Armstrong N, Nugent C, Moore G, Finlay D (2009) Mapping user needs to smartphone services for persons with chronic disease. In: Proceedings of the 7th international conference on smart homes and health telematics: ambient assistive health and wellness management in the heart of the city (ICOST 2009), Tours, France, 1–3 July 2009. Springer, pp 25–31. doi:10.1007/978-3-642-02868-7_4
19. Nachman L, Baxi A, Bhattacharya S, Darera V, Kodalapura N, Mageshkumar V, Rath S, Acharya R (2010) Jog falls: a pervasive

healthcare platform for diabetes management. In: Proceedings of pervasive 2010, Helsinki, Finland, 17–20 May 2010. pp 94–111
20. Arteaga SM, Kudeki M, Woodworth A (2009) Combating obesity trends in teenagers through persuasive mobile technology. Access Comput (ACM SIGACCESS) 94:17–25. doi:10.1145/1595061.1595064
21. Barth J, Klucken J, Kugler P, Kammerer T, Steidl R, Winkler J, Hornegger J, Eskofier B (2011) Biometric and mobile gait analysis for early diagnosis and therapy monitoring in Parkinson's disease. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society (EMBC 2011), 30 Aug–3 Sept 2011. IEEE, pp 868–871
22. Tacconi D, Mayora O, Lukowicz P, Arnrich B, Setz C, Troster G, Haring C (2008) Activity and emotion recognition to support early diagnosis of psychiatric diseases. In: Proceedings of the 2nd international conference on pervasive computing technologies for healthcare (PervasiveHealth 2008), Tampere, Finland, 30 Jan–1 Feb 2008. IEEE, pp 100–102
23. Ermes M, Parkka J, Mantyjarvi J, Korhonen I (2008) Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. IEEE Trans Inf Technol Biomed 12(1):20–26
24. Hamanaka M, Murakami Y, Kim D, Miura Y, Atsuta K, Lee S (2011) System for detecting kindergartners. Syst Cybern Inf 9(2):39–45
25. Choudhury T, Philipose M, Wyatt D, Lester J (2006) Towards activity databases: using sensors and statistical models to summarize people's lives. IEEE Data Eng Bull 29
26. Lindström M, Stahl A, Höök K, Sundström P, Laaksolathi J, Combetto M, Taylor A, Bresin R (2006) Affective diary: designing for bodily expressiveness and self-reflection. In: Paper presented at the CHI '06 extended abstracts on human factors in computing systems (CHI EA 2006), Montréal, QC, Canada, 24–27 April 2006, pp 1037–1042
27. Bajrami G, Derawi MO, Bours P (2011) Towards an automatic gait recognition system using activity recognition (wearable based). In: Proceedings of the 3rd international workshop on in security and communication networks (IWSCN 2011), Gjovik, Norway, 18–20 May 2011
28. Ofstad A, Nicholas E, Szcodronski R, Choudhury RR (2008) AAMPL: accelerometer augmented mobile phone localization. In: Proceedings of the 1st ACM international workshop on mobile entity localization and tracking in GPS-less environments (MELT 2008), San Francisco, CA, USA, 14–19 Sept 2008. ACM, pp 13–18. doi:10.1145/1410012.1410016
29. Chon J, Hojung C (2011) LifeMap: a smartphone-based context provider for location-based services. IEEE Pervasive Comput 10(2):58–67
30. Zhang S, McCullagh P, Nugent C, Zheng H (2010) Activity monitoring using a smart phone's accelerometer with hierarchical classification. In: Proceedings of the 6th international conference on intelligent environments (IE 2010) Kuala Lumpur, Malaysia, 19–21 July 2010. IEEE, pp 158–163
31. Sun L, Zhang D, Li B, Guo B, Li S (2010) Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In: Ubiquitous intelligence and computing. Lecture notes in computer science, vol 6406. Springer Berlin/Heidelberg, pp 548–562. doi:10.1007/978-3-642-16355-5_42
32. Yang J (2009) Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In: Proceedings of the 1st international workshop on interactive multimedia for consumer electronics (IMCE 2009), Beijing, China, 23 Oct 2009. ACM, pp 1–10. doi:10.1145/1631040.1631042
33. Miluzzo E, Lane ND, Fodor K, Peterson R, Lu H, Musolesi M, Eisenman SB, Zheng X, Campbell AT (2008) Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In: Proceedings of the 6th ACM conference on embedded networked sensor systems (SenSys 2008), Raleigh, NC, USA, 5–7 Nov 2008. ACM, pp 337–350. doi:10.1145/1460412.1460445
34. Khan AM, Lee YK, Lee SY, Kim TS (2010) Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In: Proceedings of the 5th international conference on future information technology (FutureTech 2010) Busan, Korea, 20–24 May 2010. IEEE, pp 1–6
35. Chipchase J, Yanquing C, Ichikawa F (2007) Where's the phone? Selected data. Survey. Nokia
36. Android-API (2009) Sensor class. http://developer.android.com/reference/android/hardware/Sensor.html. Accessed 28 Sept 2011
37. Xsens-Technologies (2008) MTi and MTx user manual and technical documentation
38. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of the 11th conference on uncertainty in artificial intelligence (UAI 1995), Montreal, QC, Canada, 18–20 Aug 1995. Morgan Kaufmann, pp 338–345
39. Kohavi R (1995) The power of decision tables. In: Proceedings of the 8th European conference on machine learning, Heraclion, Crete, Greece, 25–27 April 1995, pp 174–189
40. Quinlan R (1993) C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo, CA
41. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11(1):10–18. doi:10.1145/1656274.1656278
42. Myra LS, Witmer JA, Schaffner AA (2012) Statistics for the life sciences, 4th edn. Pearson Education
43. Nations FaAOotU (2004) Human energy requirements: report of a joint FAO/WHO/UNU expert consultation. FAO food and nutrition technical report series, Rome
44. Puyau MR, Adolph AL, Vohra FA, Zakeri I, Butte NF (2004) Prediction of activity energy expenditure using accelerometers in children. Med Sci Sports Exerc 36(9):1625–1631
45. Kurpad AV, Raj R, Maruthy KN, Vaz M (2006) A simple method of measuring total daily energy expenditure and physical activity level from the heart rate in adult men. Eur J Clin Nutr 60:32–40
46. Ainsworth BE, Haskell WL, Whitt MC, Irwin ML, Swartz AM, Strath SJ, O'Brien WL, Bassett DR Jr, Schmitz KH, Emplaincourt PO, Jacobs DR Jr, Leon AS (2000) Compendium of physical activities: an update of activity codes and MET intensities. Med Sci Sports Exerc 32(9):498–504