

# HUMAN ACTIVITY TRACKING BY MOBILE PHONES THROUGH HEBBIAN LEARNING

Jafet A. Morales and David Akopian

Department of Electrical and Computer Engineering, University of Texas at San Antonio,  
San Antonio, Texas, USA

## **ABSTRACT**

*A method for human activity recognition using mobile phones is introduced. Using the accelerometer and gyroscope typically found in modern smartphones, a system that uses the proposed method is able to recognize low level activities, including athletic exercises, with high accuracy. A Hebbian learning preprocessing stage is used to render accelerometer and gyroscope signals independent to the orientation of the smartphone inside the user's pocket. After preprocessing, a selected set of features are obtained and used for classification by a k-nearest neighbor or a multilayer perceptron. The trained algorithm achieves an accuracy of 95.3 percent when using the multilayer perceptron and tested on unknown users who are asked to perform the exercises after placing the mobile device in their pocket without any constraints on the orientation. Comparison of performance with respect to other popular methods is provided.*

## **KEYWORDS**

*accelerometer, gyroscope, human activity recognition, smartphone*

## **1. INTRODUCTION**

Human activity recognition (HAR) has been intensively studied over recent years [1]. From a broad perspective, activity monitoring can be done by processing signals from sensors external to subject's body [2], video processing [3], and sensors mounted on the subject's body [4] [5] [6]. Examples of specific study aspects are such as measurement of arm [7] and foot [8] motions and trunk postural changes [9]. Recently HAR-related research gained strong attention due to healthcare-related mobile applications. These applications are able to provide users with information related to their daily activities and exercise routines. The current paper introduces an activity tracking method that uses signals from sensors worn by the subject, particularly signals coming from accelerometers and gyroscopes, which are typically mounted on modern smartphones. The proposed algorithms, however, can possibly be used on wearable sensors that are otherwise mounted on the user. In this section a brief survey of HAR research that makes use of wearable accelerometers and gyroscopes is provided, including problem definition, architecture of the sensor array, capture and preprocessing of the data, feature set and classification, and performance evaluation.

In past research, activities that were tracked via wearable sensors ranged from *low level* to *high level* activities, which are loosely defined in the literature. In the current paper, low level activities refer to stationary or postural activities (i.e., remaining seated or standing), transitional activities (i.e., sitting or standing up), or repetitive activities that are composed of more than one short-duration cycle (i.e., performing squats as an exercise or simply walking). High level activities are much harder to detect using only accelerometer and gyroscope instrumentation, since they vary from individual to individual and there are other factors involved, such as time of the day and location. The *work* activity, for instance, is a high level activity that depends on the individual, day of the week, and location. The motion signals obtained from a teenager playing

video games at home on a Tuesday morning may actually be very similar to those of middle aged man working at the office. Thus high level activity recognition seems to be out of the scope of algorithms that only make use of accelerometer and gyroscope signals from smartphones. This also applies to many other high level activities such as *going-on-a-vacation*. The authors believe that positioning signals, time signals, and user provided data can be used by smartphones to identify high level activities. But this proposition is under the assumption that high level activities are sequences of low level activities that are somewhat similar as long as they are performed by the same individual, at the same location, and in the same seasonal timeframe. Still, most HAR research is at the current time focused on processing the signals coming out of available sensors in order to achieve better low level activity recognition. Researchers usually attempt to identify at least the most basic activities such as *walking* and *not moving*[4], [5], [6], [10], [11], [12], [13]. Some researchers push their algorithms further by making them identify a larger set of activities that includes walking, travelling the stairs up or down, or not moving [4], [5], [12]. Sports-related activities, such as remaining still, running, walking, and bicycling, have also been detected with accelerometers and gyroscopes [4], [12]. Some algorithms have been designed to distinguish amongst stationary activities such as sitting or remaining upright [4], [6], [10] [11]. In general, the contribution to low level activity recognition done by the algorithms in discussion is usually in the preprocessing, feature extraction, and classification stages they propose. In terms of accuracy, 90% levels are already achieved most of the time. The problem is not in the detection accuracy, but on the small number of activities that have been explored and on the usability of these algorithms outside of the laboratory.

Most wearable sensor based HAR algorithms, supervised, or unsupervised, are far from performing low level activity HAR with the level of accuracy that humans can achieve through a visual inspection. The ideal wearable sensor HAR algorithm would be able to recognize all of the low level activities performed by any user during an entire day. This is a task extremely easy for human beings by visual inspection, but currently impossible for smartphones. In general, human beings can perform a large number of activities throughout the day and this may be too much to ask for any contemporary machine.

There is an interrelationship between the activity set and the learning methods that can be used to identify each activity. One method that is good for detecting one set of activities (i.e., using kNN to classify walking, running, and other exercise activities) may not necessarily be the best choice for detecting another activity set (i.e., fall detection). Another issue with some studies is that outside of a lab setting, users position the smartphone with different orientations and on different on-body locations. Studies that have addressed this problem include [11] [14] [15] [16].

The placement architecture of the sensor array varies in reported research. Sensor arrays differ in the type of sensors, number of sensors, and location of sensors. In most cases, a single three-axial accelerometer is attached to the body, although in some cases more than one accelerometer have been used [4]. Up to 5 bi-axial accelerometers attached at locations in ankle, wrist, hip, arm, and thigh in [4] but it was determined that there was not much improvement in accuracy to using only hip and wrist or thigh and wrist locations. The waist location has already been studied in [4], [6], [11] and the pelvic region was studied in [5].

The way the data is captured is different for every researcher. The sampling rate is usually between 20 Hz [10], [13] and 50 Hz [5], [11]. Most researchers do not specify why they use a particular sample rate. The size of the window to be processed ranges from one second [6], [11], [12] up to ten seconds [10]. Windows that overlap at 50% were used in [4], [5] but no explanation is given about the benefit of using overlapping windows and whether this benefit depends on the period of the activity that is being monitored and the size of the window. Overlapping windows of 50% may be better for online recognition in the sense that they can provide the user with a classification result in intervals of time that are half the size of those of non-overlapping window

systems. But they can also increase computational expense by mobile phones and increase the load on the battery as the calculations are performed online by the device.

Motion signals are usually preprocessed before performing classification. For example, spikes in the three axes of inertial sensors can be removed by using an FIR filter of fifth order with a cut-off close to .5 radians/sec or a similar filter, as done in [13]. A median filter can also be used, as done in [6]. Gravity can then be subtracted by additional preprocessing [6], [11]. Gravity was eliminated in [6] with a .25Hz cutoff elliptic low pass IIR filter of third-order. In [11] the signal is normalized by subtracting mean and dividing by standard deviation.

Several features have been used for activity recognition with wearable motion sensors. Frequency domain features are in some cases obtained by finding the location of peaks [6], or taking discrete fourier transform (DFT) coefficient values or DFT bin values [10], [12]. However, most popular features are obtained from the time domain. These include mean acceleration [4], [5], [10], [11], power [4], [5], [12], standard deviation [5], [10], [11], periods [10], [12], [13], and signal magnitude areas [6]. All of these features can be calculated individually for each channel or for the magnitude of the 3-dimensional acceleration vector. But features that relate different channels, such as interchannel correlation have also been used [4], [5]. For classification, decision trees are popular [4], [5], [6], [10], [13], along with kNN [4], [5], [11], SVMs [5], [8], neural networks [10], dynamic time warping (DWT) [12], and plurality voting [5].

Currently, researchers obtain accuracies usually between 90 and 95% [6], [10], [11], [12] for HAR using wearable motion sensors. Accuracy greater than 99% has been reported in [13] for distinguishing between the states of walking and remaining idle. There are two popular ways of testing the accuracy of human activity recognition algorithms based on wearable sensor signals. When doing the personalized validation, the algorithm is trained and tested on each individual separately by using all frames of the individual in question but one. The algorithm is then tested on the remaining frame of that individual [4], [5], [12]. When performing the leave-one-subject-out type of validation the method in question is first trained on all subjects but one. The method is then tested on the remaining subject. Accuracy is typically higher in the personalized validation but the leave-one-user-out type of validation is also important because it is an indicator of potential for deployment to the masses. In some cases all frames from all users and all activities are mixed together into a pool and the system is tested by performing a folded cross-validation on a percentage of frames that are randomly chosen from the pool [5], [10], [12]. Most of the methods in question can classify the commonly practiced cyclic activities of jogging, walking, bicycling, going up or going down the stairs. For this type of activities, simple features such as the location and height of a peak in the frequency domain, the angle with respect to gravity, or a measurable period are not enough information to classify the activities. In such cases testing by cross-validation is possible because a data can also be used for training.

Not all algorithms of interest can be tested using cross-validation. In some cases, accuracy is calculated as the portion of frames in the dataset which are correctly classified [6], [10]. This is the case of algorithms that make decisions based on a small set of equations that are parameterized by just a few features, rather than by using classifiers trained on previous examples. The benefit of this type of algorithms is that they do not need to be trained and therefore are not overfitted to the dataset of the researcher. Also, these algorithms are not complex and processing intensive. However, they can only be used to distinguish simple activities, such as certain stationary positions, transitional activities, and being active or inactive. For example, setting a threshold for the angle between a horizontal axis (alternatively, an axis in the direction of gravity) and an axis that goes through the device from top to bottom can be used to distinguish between sitting and standing up. The method proposed in [6] can be used to identify the activities of falling, sitting, standing, being active while standing up, being active while lying, and lying on back, front, left, or right side of the body. The algorithm uses a decision

thee that makes decisions based on thresholding of signal magnitude area, acceleration peaks, and an angle with respect to gravity. Validation based on the percentage of frames that are classified correctly without performing data partitioning for training and testing also applies to the algorithm proposed in [13]. In this case, the algorithm can discriminate between resting and walking by calculating an Average Magnitude Difference Function (AMDF) at a given delay and finding the location of peaks. In general, the performance of wearable inertial sensor-based HAR algorithms is usually shown in accuracy percentages and confusion tables.

There are some similarities between attachable motion sensor and smartphone based methods for HAR. First of all, the type of signals are typically accelerometer and gyroscope signals. Parameters for signal capture, such as the frame size and the rate of sampling can also be expected to be similar. Calculated features and methods of classification may also be similar between smartphone based methods and attachable motion sensor methods, even though sensor may be worn differently. Using a mobile phone typically means that the sensors will be attached to the arm or to the waist or inside a pocket. Already explored pockets include front pant pockets [10], [11], [12], shirt pockets [11], and jacket pockets [13]. When the inertial sensors or the smartphone device are attached to a body part, the orientation of the signals in the training and testing stages will be the same. In this case, high accuracy results can be expected. This situation is mostly seen in users who do not want the device jumping around through the empty space in their pockets while they exercise and at the same time do not want to hold it in their hands. Outside of these particular scenarios, the user can be expected to have the device in his hand(s), pockets, purse, or laying on a table or other flat surface. The algorithm proposed in this paper is tested in front pant pockets, but the authors expect decent performance on fixed scenarios as well, due to pocket signals being more unpredictable than fixed device signals. Pocket signals require more processing than fixed device signals in order to account for many additional factors. First, the device can be oriented on different ways as it is placed by the user on his/her pocket. Pocket signals are also different because the device bounces around in the free space of the pocket. Finally, the pocket deforms and moves on the surface of the body depending on the location of the pocket, clothing, and shape of the body. None of these effects are present in fixed orientation signals, where effects on variability are only due to the angle at which the device is attached around the limb, how far it is attached from the joint, and the size of the individual.

Section 2 of this paper proposes a window-based as well as an adaptive method for obtaining orientation-invariant accelerometer and gyroscope signals. The features used by a proposed HAR algorithm that uses orientation-invariant signals and the method used to generate this list of features are described in Section 3. The classification stage of the HAR algorithm is described in Section 4 in a brief manner. Section 5 of the paper focuses on testing and performance of the HAR algorithm. Finally, Section 6 presents conclusions on the overall performance of the algorithm.

## **2. ORIENTATION-INVARIANT ACCELEROMETER AND GYROSCOPE SIGNALS**

Currently most methods for HAR are based on wearable inertial sensors calculate features directly from the acceleration in the X, Y, and Z directions of the device's coordinate system. This means that, for those methods to give maximum accuracy, the device must be oriented the same way during the training and testing phases. In scenarios outside of the laboratory, however, the orientation of a smartphone with respect to the user's frame is less predictable. One way to eliminate the effect of random orientations is to process the signal in such a way that a particular activity always generates the same processed signal, but still different signals are obtained for different activities. This can be done by transforming the signal to a different coordinate system. This coordinate system may or may not be the same for all activities. A separate coordinate system can be used for each activity, as long as the resulting acceleration signatures are not the same. Principal component analysis (PCA) was used for preprocessing the motion signals in

[11], [14], [15], [16]. Performance is to be evaluated differently from smartphone based methods to attachable accelerometer based methods. If the algorithm is to work regardless of the orientation of the accelerometers, performance must be evaluated by testing on several orientations. In such case, it can be evaluated by training on a single orientation and testing on several orientations as done in [11], or simply by training and testing on randomly chosen orientations, as done in [16].

A PCA was used in [11] to classify activities recorded with a mobile phone with a set of different orientations. Gravity was first eliminated by subtracting mean. This results in a signal that contains only linear acceleration, that is, the acceleration due to motion. The acceleration along the direction of the subtracted gravity signal is then used as an orientation-invariant component. An additional orientation-invariant component is then obtained by finding the principal component of acceleration on the plane orthogonal to the gravity vector. The third orientation-invariant signal is then found by taking the acceleration orthogonal to the other two components. A different approach to orientation-invariance was taken in [14], [15], where the PCA transform of acceleration was calculated in three-dimensions, and not only on the two dimensions orthogonal to gravity. Only the principal component of acceleration was used to extract features from in [14][15]. Three PCA components of accelerometer and three PCA components of gyroscope were used in [16] to distinguish human activities by using a smartphone.

One of the disadvantages of using the method described above to obtain orientation-invariant signals is that the transform operation must be performed in an entire window in the vicinity of the sample of interest, so that transformation of the signal is typically performed in chunks that may or may not overlap, in order to reduce computational complexity. This introduces a minimum delay due to the spacing between the last sample of one window and the next one. Also, because the PCA transform is very signal dependent there may be “blind spots” in which part of the window corresponds to one activity and another part of the window corresponds to a different activity. In such cases, even though the PCA transform has provided a fully recoverable orientation-invariant signal, the transformed signal does not correspond to any of the stored activities, making it impossible to target the activity at the sample of interest.

Using the generalized Hebbian algorithm, orientation-invariant components for a motion sample can be calculated as soon as the 3-dimensional motion sample is received, without having to perform a PCA on an entire window of data around the sample of interest. This eliminates the lower boundary on the delay due to spacing between windows and delegates it to the feature extraction algorithm. Additionally, blind spots can be detected as changes in the transform operator. Once a blind spot has been spotted, the results of classification can be ignored in order to avoid delivering faulty results, or a different classification algorithm may be used to target the blind spot.

## **2.1. ORIENTATION-INVARIANT WINDOWS BY PRINCIPAL COMPONENT ANALYSIS**

One way of obtaining a set of orientation-invariant motion signals is by applying principal component analysis (PCA) to the original accelerometer and gyroscope signals. The resulting six motion signal components can then be used for human activity recognition, as done in [16] and explained only for the sake of completeness in this paper. When using PCA transform signals, the orientation of the PCA coordinates with respect to the smartphone and the user will depend on how variance in the 3-dimensional motion signals is distributed with respect to the smartphone and how the smartphone is positioned with respect to the user, making the coordinate system extremely case dependent. It is unlikely that two different activities of interest from a small set of activities will generate similar motion signatures. In this section, application of PCA to wearable inertial sensor 3-dimensional signals is described in detail. For a more in-depth discussion of PCA analysis on signals of even higher dimensionality the reader is referred to [17].

A sample of an acceleration frame is represented by

$$\mathbf{a}^j = \begin{bmatrix} a_1^j \\ a_2^j \\ a_3^j \end{bmatrix} \quad (1)$$

where  $a_i^j$  is the  $j$ th acceleration sample in the  $i$ th coordinate. A window of acceleration samples can then be aggregated into a matrix

$$\mathbf{A} = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^n \\ a_2^1 & a_2^2 & \cdots & a_2^n \\ a_3^1 & a_3^2 & \cdots & a_3^n \end{bmatrix} \quad (2)$$

where each row contains the samples for acceleration along one axis while each column is an acceleration vector sample. The PCA transform is applied on a centralized matrix that is calculated by subtracting the average acceleration vector from each column

$$\bar{\mathbf{A}} = \begin{bmatrix} -1 & -2 & \cdots & -n \\ a_1 & a_1 & \cdots & a_1 \\ -1 & -2 & \cdots & -n \\ a_2 & a_2 & \cdots & a_2 \\ -1 & -2 & \cdots & -n \\ a_3 & a_3 & \cdots & a_3 \end{bmatrix} \quad (3)$$

An orthonormal transform can be used to decorrelate the dimensions of  $\bar{\mathbf{A}}$  through the linear transformation

$$\bar{\mathbf{B}} = \mathbf{P}_{\bar{\mathbf{A}}} \bar{\mathbf{A}} \quad (4)$$

where  $\mathbf{P}_{\bar{\mathbf{A}}}$  is the transform operator for the window  $\bar{\mathbf{A}}$ . This transform is found by performing a diagonalization of the covariance matrix of  $\bar{\mathbf{A}}$ . The covariance matrix of  $\bar{\mathbf{A}}$  is calculated as

$$\mathbf{\Omega}_{\bar{\mathbf{A}}} = \frac{1}{n} \bar{\mathbf{A}} \bar{\mathbf{A}}^T \quad (5)$$

While  $\mathbf{\Omega}_{\bar{\mathbf{A}}}$  is not a diagonal matrix, the covariance matrix of  $\bar{\mathbf{B}}$ , denoted as  $\mathbf{\Omega}_{\bar{\mathbf{B}}}$ , should be a diagonal matrix, meaning that the components of  $\bar{\mathbf{B}}$  are decorrelated. Since  $\mathbf{\Omega}_{\bar{\mathbf{A}}}$  is symmetric, it is diagonalized with

$$\mathbf{\Omega}_{\bar{\mathbf{A}}} = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^T \quad (6)$$

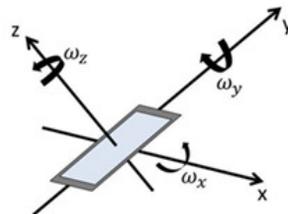


Fig. 1. Coordinate System for the Smartphone used in this Project

where  $\Lambda$  is a diagonal matrix with eigenvalues for  $\bar{\mathbf{A}}$ , and  $\Phi$  is the orthonormal eigenvector matrix of  $\bar{\mathbf{A}}$ . The PCA transform is then

$$\mathbf{P}_{\bar{\mathbf{A}}} = \Phi^T \quad (7)$$

Once the signal has been preprocessed for orientation invariance, HAR can be done by using the pattern recognition techniques described before, so long as the motion detecting device (i.e., smartphone) is placed in the same location during training and testing. Similar preprocessing can also be used to obtain orientation-invariant angular velocity signals. Three-axis gyroscopes output three signals. Each signal is the angular speed in rad/sec about an orthogonal axis. A typical coordinate system for smartphone motion signals is portrayed in Figure 1. In smartphones, the axes for accelerometers and gyroscopes are oriented the same way.

Angular frequency can be represented with a vector

$$\mathbf{w} = \mathbf{u}w \quad (8)$$

where

$$\mathbf{u} = \frac{1}{\sqrt{w_x^2 + w_y^2 + w_z^2}} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (9)$$

is a unit vector pointing in the direction of the axis of rotation and

$$w = \sqrt{w_x^2 + w_y^2 + w_z^2} \quad (10)$$

where  $w_i$  is the angular frequency or speed about the  $i$  axis, and  $w$  is the angular speed around the axis of rotation of the angular velocity vector. Angular velocity vectors can also be made orientation-invariant by a linear transformation. In this case, the three signals obtained from the three-axial gyroscope are PCA transformed.

Many activities have distinguishing patterns in more than one of the six components that can be obtained through the transformations above. This can easily be understood by picturing a user jogging at a constant speed with the smartphone in his front right pocket. In this activity, the mobile phone moves back and forth in the direction the user is running, but also up and down as each foot jumps and falls back on the floor. The smartphone does not move along a single axis, so that more than one axis is needed to represent it. Therefore, using only the principal component from the PCA coordinate system may not be enough to discriminate between different activities in some cases. Similarly, angular velocity signals coming from gyroscopes are not necessarily well represented using only the principal component. When a user is jogging with a mobile phone in one of his/her front pant or short pockets, clothes can force the device to orbit around the user's leg, always facing the outside of the leg. This means the device will rotate about an axis that goes through it and that is parallel to the leg of the user. At the same time, due to the typical way humans move their legs when they run, the device will also orbit around a line going through the user's waist, perpendicular to gravity and the direction of running. Therefore, the principal component may not represent the rotations of the smartphone well enough to allow for accurate classification when there are several activities to be classified. Example PCA-transformed signals for the *jogging* activity are shown in Figure 2. In some activities, patterns can also be detected visually in the third component of the PCA transformation.

## 2.2. ORIENTATION-INVARIANT SAMPLES BY GENERALIZED HEBBIAN ALGORITHM

The generalized Hebbian algorithm is an unsupervised learning method based on Hebb's learning rule about the way synaptic strengths in the brain are adjusted. According to Hebb's learning rule, strengths of the synapses in the brain are adjusted through changes that are proportional to the pre-synaptic input and the output in the post-synaptic neuron, which means that the rate of adjustment is proportional to the correlation between the input and the output signals. This idea is used in the generalized Hebbian algorithm in order to decorrelate the components in a multi-dimensional signal.

The simplified neuron model is given by

$$y = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^n w_i x_i \quad (11)$$

where  $w$  is a synaptic weight. Hebb's learning rule for neurons can be written as follows

$$\Delta w_i = \alpha x_i y \quad i = 1, \dots, n \quad (12)$$

where  $\Delta w_i$  is the change in the weight for  $x_i$ . The weight update for every iteration can be written as

$$w_i(n+1) = w_i(n) + \alpha x_i y \quad (13)$$

Normalizing to ensure that the length of the weight vector is one so that the vector does not grow infinitely large,

$$w_i^{t+1} = \frac{w_i^t + \alpha x_i y}{\left( \sum_{m=1}^n (w_m^t + \alpha x_m y)^2 \right)^{1/2}} \quad (14)$$

where  $r = 2$  for Euclidean distance. Using a Taylor expansion, the expression for  $w_i(t+1)$  can be simplified

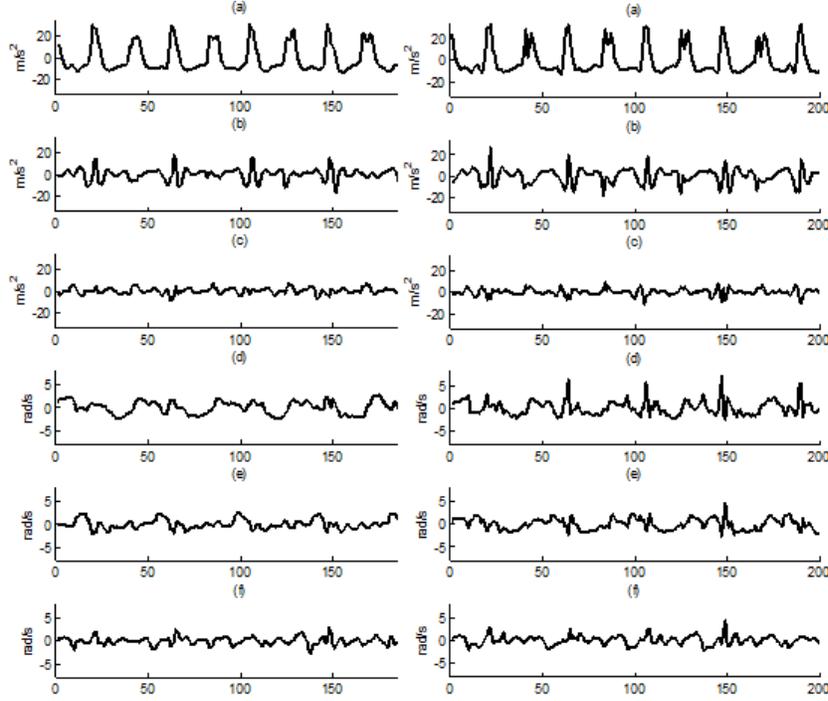


Fig. 2. Principal components (left column) and generalized Hebbian components (right column) of motion signals for jogging activity; first component of acceleration (a); second component of acceleration (b); third component of acceleration (c); principal component of gyroscope signal (d); second component of gyroscope signal (e); third component of gyroscope signal (f).

$$w_i^{t+1} \approx w_i^{t+1}(\alpha' = 0) + \alpha \left. \frac{\partial w_i^{t+1}(\alpha')}{\partial \alpha'} \right|_{\alpha'=0}$$

$$= w_i^{t+1}(\alpha' = 0) + \alpha \left( \frac{yx_i \sqrt{\sum_{j=1}^n (w_j^t + \alpha' yx_j)^2} - (w_i^t + \alpha' yx_i) \frac{\sum_j yx_j (w_j^t + \alpha' yx_j)}{\sqrt{\sum_j (w_j^t + \alpha' yx_j)^2}}}{\sum_j (w_j^t + \alpha' yx_j)^2} \right) \Bigg|_{\alpha'=0} \quad (15)$$

Substituting  $\sum_j (w_j^t)^2 = 1$  in the expression above,

$$w_i^{t+1}(\alpha) \approx w_i^t + \alpha (yx_i - w_i^t \sum_j yx_j w_j^t) \quad (16)$$

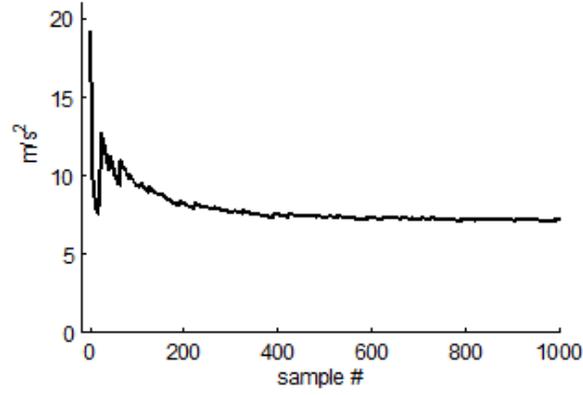


Fig. 3. Convergence of the Hebbian algorithm to the jogging activity principal components can be verified by observing the difference between the true 3-dimensional acceleration  $\sqrt{a_x^2 + a_y^2 + a_z^2}$  and the acceleration obtained by inverting the transformation on the orientation-invariant signals

The update for the weight can then be rewritten as

$$\Delta w_i(\alpha) = \alpha(yx_i - w_i y \sum_j x_j w_j) \quad (17)$$

This result is Oja's rule and is typically written in vector form as

$$\Delta \mathbf{w} = \alpha(\mathbf{x}y - y^2 \mathbf{w}) \quad (18)$$

Oja's rule is related to the principal components. Substituting  $y = \mathbf{x}^T \mathbf{w}$ ,

$$\Delta \mathbf{w} = \alpha(\mathbf{x}\mathbf{x}^T \mathbf{w} - \mathbf{w}^T \mathbf{x}\mathbf{x}^T \mathbf{w}) \quad (19)$$

Upon convergence,  $\mathbf{w}$  is approximately constant and  $\Delta \mathbf{w} \approx 0$  so that the above equation can be rewritten as

$$\mathbf{x}\mathbf{x}^T \mathbf{w} \approx \mathbf{w}^T \mathbf{x}\mathbf{x}^T \mathbf{w} \quad (20)$$

For the average  $\mathbf{x}\mathbf{x}^T$ , the above equation can be written as

$$E[\mathbf{x}\mathbf{x}^T] \mathbf{w} \approx \mathbf{w}^T E[\mathbf{x}\mathbf{x}^T] \mathbf{w}$$

$$\Omega_x \mathbf{w} \approx \mathbf{w}^T \Omega_x \mathbf{w} \quad (21)$$

which is the eigenvector, eigenvalue equation for  $\Omega_x$ , which is the covariance matrix for  $\mathbf{x}$ . In general, the eigenvectors with the largest eigenvalues point in the direction of highest variance. For a network with a single output neuron, only one eigenvector is defined by  $\mathbf{w}$ . This eigenvector  $\mathbf{w}$  points in the direction of the principal component. Therefore, for a single output neuron that learns via Oja's rule (18), it can be said that upon convergence,  $\mathbf{w}^T$  is the PCA

Table 1. Table of Features Obtained from Hebbian Signals that were Selected by CFS.

Name	Calculation	$a'_1$	$a'_2$	$a'_3$	$w'_1$	$w'_2$	$w'_3$
Mean	$\bar{x} = \frac{1}{N} \sum_{n=0}^N x[n]$	*	*	*	*	*	*
Normalized cross-correlation	$R_{xy} = \frac{1}{N} \sum_{n=0}^N (x[n] - \bar{x})(y[n] - \bar{y}) / (\sigma_x \sigma_y)$	*		*	*	*	
Standard deviation	$\sigma$	*			*		
Normalized Signal Magnitude Area	$\bar{x} = \frac{1}{N} \sum_{n=0}^N  x[n] $	*				*	
Power	$P = \frac{1}{N} \sum_{n=0}^N x^2[n]$	*		*			
Power contained in [10Hz to 20Hz]	$P_{x_{f_1, f_2}} = \sum_{k=f_1}^{f_2}  X[k] ^2$ $K_1 = \frac{f_1}{f_s/2} N, K_2 = \frac{f_2}{f_s/2} N$ $f_1 = 10\text{Hz}, f_2 = 20\text{Hz}$		*				*
Entropy	$H(x[n])$	*	*	*	*	*	*
Repetitions per second	$\frac{1}{M} \sum_{n=0}^{M-1} \left( \frac{1}{N} \sum_{m=0}^{N-1} x[n+m] y[m] \right) / \sigma_x \sigma_y$	*					

transform operator and  $\mathbf{w}^T \mathbf{x}$  is the principal component [18]. Multiple component extraction methods are explored in [19].

In this paper, the proposed method for HAR makes use of orientation-invariant signals obtained through the generalized Hebbian algorithm. Figure 2 shows the motion signals generated by a user as he is jogging. Figure 3 shows the difference between the total acceleration signal  $\sqrt{a_x^2 + a_y^2 + a_z^2}$  and the signal recovered by inverting the transformation. Such curve, and the similarity between Hebbian components and PCA components can be used to verify convergence of the Hebbian algorithm

### 3. FEATURE EXTRACTION AND CLASSIFICATION

The feature set used by the system was created by selecting from a pool of features from [5][10] and [16]. In order to improve accuracy and reduce computational expenses, a feature selection was performed. Selection was done with a correlation-based feature selection (CFS) method proposed in [20], in conjunction with a backwards greedy search. The backwards greedy search starts by using all features and removes one at each iteration step. At each iteration, the feature to be eliminated from the selection is the one for which a figure of merit is maximized when the features is not included in the subset. The purpose of the CFS feature selection algorithm is to find a subset of features that are lowly correlated with each other, but highly correlated with the class label.

Features are eliminated at each iteration until no more eliminations can be done without diminishing performance. A figure of merit from [21] is used,

$$M_s = \frac{\overline{kr}_{cf}}{\sqrt{k + k(k+1)r_{ff}}} \tag{22}$$

where  $\overline{r_{ff}}$  is the average Pearson correlation between each possible pair of features and  $\overline{r_{cf}}$  is the average Pearson correlation between each feature and the class. The features in use are not necessarily continuous. In cases where all the features in use are continuous, Pearson's correlation can be used to obtain a figure of merit in (22).

However, in order to include categorical features and classes, which are also categorical, all continuous features are discretized using a discretization algorithm in [22]. Symmetrical uncertainty is then calculated between each pair of features and used instead of Pearson's correlation in (11). Such quantity has a value in the range [0,1]. Symmetrical uncertainty is calculated as

$$U = 2 - \frac{G}{H(Y) + H(X)} \quad (23)$$

where  $H$  is information entropy and  $G = H(Y) - H(Y|X)$  is the mutual information or expected information gain for a given feature pair  $(X,Y)$ . Information gain is the decrement in the entropy of  $X$  when the value of feature  $Y$  is known, or viceversa. As opposed to information gain, symmetrical uncertainty is a normalized quantity that is not biased by the number of categories in a feature.

The feature selection can be performed by adding or eliminating features one at a time or by testing several subsets by feeding each into the classifier and checking accuracy of results. But such method is computationally expensive. In such case, the resulting feature subset would also depend on the classifier. Using a performance metric, such as the one in (11) to calculate for each subset without actually having to feed it into the classifier is less computationally expensive.

In general, the resulting feature subset will not be optimal but it will be a local minimum that produces a smaller set of features and results in better accuracy, found with a reasonable amount of computational expense. The original pool of features was made of features chosen from those described in Section I and additional features proposed by the authors. The final set of features obtained after performing feature selection in the proposed algorithm are shown in Table 1. All the features obtained during the training stage were converted to z-scores.

The authors have chosen to use kNN because there are many implementations readily available. In this case the authors have used the WEKA implementation [23]. A disadvantage of instance-based algorithms is that they are computationally expensive. However, one benefit of instance-based algorithms is their ability to adapt to new data. New instances can be added to the training data simply by calculating features vectors and storing them, as opposed to other algorithms that may require the entire classification model to be calculated again, such as decision trees. Old instances can be eliminated easily as well for adaptation. Algorithms with such ability may prove advantageous when deploying to the masses, given that each user has his or her own way of performing an activity (i.e., different users may have a different gait). A multilayer perceptron was also chosen in our implementation because of the ability of neural networks to approximate any continuous function within an input interval to any desired level of accuracy. The approximation that results for discontinuous functions can still be useful. A neural network with a single hidden layer can approximate any function by adding more neurons to the hidden layer in order to increase accuracy. Neural networks are popular for solving pattern recognitions problems and there are also many implementations readily available. In this case the authors have chosen to use the WEKA implementation. SVM was not chosen for this study because it is computationally expensive. Decision trees were not chosen either to avoid overfitting due to the high number of features in use.

Table 2. Simulation Accuracies for 236 Activity Samples (%)

Simulation of [5]			
Single Subject Test		Unknown Subject Test	
C4.5	kNN	C4.5	kNN
90.0	94.2	78.8	74.6
Simulation of [10]			
Single Subject Test		Unknown Subject Test	
C4.5	MP	C4.5	MP
81.1	92.6	82.6	78.4
Using Hebbian Signals			
Single Subject Test		Unknown Subject Test	
KNN	MP	KNN	MP
95.1	98.3	85.6	95.3

Euclidean distance was used with kNN. The highest accuracy was obtained using only one neighbor. For the multilayer perceptron, nodes with a sigmoid activation function were used. A momentum of .2 and a learning rate of .3 were used for training by backpropagation. The learning rate controls the size of the adjustments to the weights in the neural network as it is being trained by backpropagation. The momentum is a number in the range [0,1] that controls the rate of change in the adjustments. If the momentum is set to zero, the adjustments depend only on the learning rate, the error in the output of the neuron, and the value of the input, but not on the size of previous adjustments. If the momentum is set to one, the size of the adjustments will be constant. The network was trained by backpropagation for 500 epochs. A single epoch is one pass through all patterns in the training set.

#### 4. TESTING

The proposed algorithm extracts the features in Table 1 from orientation-invariant signals obtained through the generalized Hebbian algorithm and classifies them either with a kNN classifier or a multilayer perceptron. The results for the multilayer perceptron are more accurate than those obtained when using kNN. Nevertheless, kNN results are also described here for completeness and because kNN is easier to implement than the multilayer perceptron. The algorithm was tested on exercise activities from university students. The athletic exercises in the database include resting, walking, jogging, squatting, and doing jumping jacks. Each user was asked to place the smartphone in the front right pocket. The users were not asked to place the smartphone with any particular orientation inside the pocket. Each exercise frame lasts for a total of 5 seconds. A total of 236 frames were captured.

The data was captured with a smartphone that houses an InvenSense MPU-6050 three-axial accelerometer and three-axial gyro sensor. The device also includes a Digital Motion Processor (DMP) which can be used to process the signals. The DMP can also be used to process signals coming from an external sensor, such as a magnetometer that can be interfaced through I2C. The MPU-6050 transfers the data to the host also via I2C. It outputs signals with a resolution of 16-bit and can be set to span four different ranges. The MPU-6050 has a printout of 4x4x.9mm. The smartphone used for this research does not take advantage of all the capabilities of the MPU-6050, but there are boards in the market that do

Table 3. Confusion table for the Unknown Subject Test Applied to the Proposed Method.

		Predicted Class				
		Standing Up	Walking	Jogging	Jumping Jacks	Squats
Actual Class	Standing Up	50	0	0	0	4
	Walking	0	37	3	1	4
	Jogging	0	5	35	5	0
	Jumping Jacks	0	0	10	36	0
	Squats	7	6	6	0	27

Two different tests were performed: *unknown-subject* and *single-subject*. The unknown-subject test was done by training the system with all subjects except the unknown subject and then testing it on the frames from the subject on which the system was not trained. The overall accuracy of the unknown-subject was calculated by averaging accuracy across all subjects. The single-subject test, which evaluates performance when the algorithm is trained and tested in the same subject, was done by taking the samples from one subject and doing a 5-fold stratified cross-validation. The overall accuracy of the single-subject test was found by averaging accuracy across all folds and across all subjects.

A performance comparison was done with the methods in [5] and [10]. Both methods were simulated using the Weka toolkit in the classification stage. The algorithm in [5] has its own feature set and classifies instances with C4.5 and kNN. The method proposed in [10] also uses the Weka toolkit. This method was simulated by extracting the features in [10] from the acceleration window without applying orientation-invariant preprocessing, and then classifying with the multilayer perceptron and the C4.5 [24] classifiers in the Weka toolkit, using the settings that were also used for the proposed algorithm. No feature standardization procedure is mentioned in [10]. The results are shown in Table 2.

The proposed method outperforms the other two popular methods, which can easily be replicated in a lab setting and used as benchmarks. For the unknown-subject test, the proposed method performs much better than the other two methods, especially when using the multilayer perceptron. Performance improvement is expected given that different users position the smartphone in different orientations inside their pockets. A single user, however, may or may not be inclined to place the device in one particular orientation.

The high accuracy obtained with the proposed algorithm may also be due to feature standardization. For the unknown subject test the proposed algorithm obtained much higher results than the methods in [5] and [10]. For single subject activity recognition both classifiers yield accuracy higher than 95%. The authors propose using orientation-invariant preprocessing, the standardized features in Table 1, and a multilayer perceptron for classification. Yet, classification with kNN was also tested because it may be easier to implement for some of the readers. The confusion table for the test with unknown subjects with the proposed HAR algorithm is shown in Table 3.

## 5. CONCLUSION

The proposed method for human activity recognition using wearable motion sensors properly accounts for random orientations of the sensors. The method is particularly useful in the case of a smartphone inside a user's front pant pocket. The preprocessing stage of the proposed method renders the motion signals orientation-invariant. A small standardized feature set calculated from motion signals is then fed into a multilayer perceptron. The method introduced in this paper

outperforms other modern methods when classifying several low level activities from unknown subjects, including exercises typically performed by athletes.

## REFERENCES

- [1] D. Guan, T. Ma, W. Yuan, Y.K. Lee, and A.M. Jehad Sarkar, "Review of Sensor-based Activity Recognition Systems," *IETE Technical Review*, Vol. 28 Issue 5, p418, Sep/Oct (2011).
- [2] S.M. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural Pattern Identification in a Smart Home Using Binary Similarity and Dissimilarity Measures," *Proceedings of the 2011 Seventh International Conference on Intelligent Environments – IE'11*, pp.55-60, 25-28 July (2011).
- [3] S.K. Tasoulis, C.N. Doukas, I. Maglogiannis, V.P. Plagianakos, "Statistical data mining of streaming motion data for fall detection in assistive environments," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC*, pp.3720,3723, Aug. 30-Sept. 3 (2011).
- [4] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of Pervasive 2004*, vol. LNCS 3001, A. Ferscha and F. Mattern, Eds. Berlin Heidelberg: Springer-Verlag, pp. 1-17, (2004).
- [5] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, 2005. Activity recognition from accelerometer data. *Proceedings of the 17th conference on Innovative applications of artificial intelligence*, Bruce Porter (Ed.), Vol. 3. (2005).
- [6] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol.10, no.1, pp.156,167, January (2006)
- [7] G.X. Lee, K.S. Low, and T. Taher, "Unrestrained measurement of arm motion based on wearable wireless sensor network," *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No.5, pp. 1309-1317, (2010).
- [8] X. Yun, J. Calusdian, E. R. Bachmann, and R.B. McGhee, "Estimation of human foot motion during normal walking using inertial and magnetic sensor measurement," *IEEE Transactions on Instrumentation and Measurement*, Vol. 61, No.7, pp. 2059-2072, (2012).
- [9] W.-Y. Wong and M.-S. Wong, "Measurement of postural change in trunk movements using three sensor modules," *IEEE Transactions on Instrumentation and Measurement*, Vol. 58, No.8, pp. 2737-2742, (2009).
- [10] J.R. Kwapisz, G.M. Weiss, and S.A. Moore, 2011. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, Volume 12 Issue 2, Pages 74-82, December (2010).
- [11] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate Activity Recognition Using a Mobile Phone Regardless of Device Orientation and Location," *International Conference on Body Sensor Networks (BSN)*, pp.41-46, 23-25 May (2011).
- [12] V.Q. Viet, H.M. Thang, and D.J. Choi, "Balancing Precision and Battery Drain in Activity Recognition on Mobile Phone," *International Conference on Parallel and Distributed Systems (ICPADS)*, pp.712-713, 17-19 Dec. (2012).
- [13] M. Hynes, H. Wang and L. Kilmartin, "Off-the-shelf mobile handset environments for deploying accelerometer based gait and activity analysis algorithms," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp.5187,5190, 3-6 Sept. (2009).
- [14] T. Mashita, D. Komaki, M. Iwata, K. Shimatani, H. Miyamoto, T. Hara, K. Kiyokawa, H. Takemura, and S. Nishio, "A content search system for mobile devices based on user context recognition," *Proceedings of IEEE Virtual Reality*, Pages 1-4 (2012).
- [15] T. Mashita, K. Shimatani, M. Iwata, H. Miyamoto, D. Komaki, T. Hara, K. Kiyokawa, H. Takemura, and S. Nishio, "Human activity recognition for a content search system considering situations of smartphone users," *Proceedings of IEEE Virtual Reality*, Pages 1,2, 4-8 March (2012).
- [16] J.A. Morales, D. Akopian, S. Agaian, "Human activity recognition by smartphones regardless of device orientation", *Proceedings of SPIE Conference on Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications*, Reiner Creutzburg and David Akopian, Editors, 18 February (2014).
- [17] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, Volume 43, Issue 4, Pages 1531-1549, April (2010).

- [18] E. Oja, (1982) A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, Vol. 15, No. 3, pp. 267-273, 1 November (1982).
- [19] S. Becker and M. Plumbley, "Unsupervised Neural Network Learning Procedures for Feature Extraction and Classification", *Applied Intelligence*, Vol. 6, No. 3, pp. 185-203 (1996).
- [20] M.A. Hall and L.A. Smith, Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper, *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, Pages 235-239 (1999).
- [21] E.E. Ghiselli, *Theory of Psychological Measurement*. McGraw-Hill, (1964).
- [22] U.M. Fayyad and K.B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *Proceedings of the International Joint Conference on Uncertainty in AI*, pp. 1022-1027, 1 September (1993).
- [23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1 (2009).
- [24] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©1993.
- [25] K. Hwang and S.Y. Lee, "Environmental audio scene and activity recognition through mobile-based crowdsourcing," *IEEE Transactions on Consumer Electronics*, vol.58, no.2, pp.700-705, May (2012).

## AUTHOR

**Jafet Morales** is a Ph.D. graduate from the University of Texas at San Antonio (UTSA) in the area of digital signal processing. He received his M.S. degree in computer engineering from St. Mary's University at San Antonio, Texas, where he wrote a thesis on fundus image quality assessment. He has eight publications in signal processing, computer science, and physics. His current interests include machine learning and virtual reality.



**David Akopian** is a professor at the University of Texas at San Antonio (UTSA). Prior to joining UTSA, he was a Senior Research Engineer and Specialist with Nokia Corporation from 1999 to 2003. From 1993 to 1999 he was a member of the teaching and research staff of Tampere University of Technology, Finland, where he also received a Ph.D. degree in electrical engineering. Dr. Akopian's current research interests include digital signal processing algorithms for communication, positioning and navigation algorithms, and dedicated hardware architectures and platforms for healthcare applications. He authored and co-authored more than 35 patents and 140 publications. He served in organizing and program committees of many IEEE conferences and co-chairs SPIE's annual Multimedia on Mobile Devices conference. His research has been supported by National Science Foundation, National Institutes of Health, U.S. Air Force, U.S. Navy, and Texas Higher Education Coordinating Board.

