

An evolving machine learning method for human activity recognition systems

Javier Andreu · Plamen Angelov

Received: 4 June 2011 / Accepted: 13 September 2011 / Published online: 5 October 2011
© Springer-Verlag 2011

Abstract In this paper is presented a novel approach for human activity recognition (HAR) through complex data provided from wearable sensors. This approach considers the development of a more realistic system which takes into account the diversity of the population. It aims to define a general HAR model for any type of individuals. To achieve this much-needed processing capacity, this novel approach makes use of customizable, self-adaptive, self-development capacities of the so-called machine learning technique named evolving intelligent systems. An online pre-processing model to suit real-time capacities has been developed and is also explained in detail in this paper. Additionally, this paper provides valuable information on sensor analysis, online feature extraction, and evolving classifiers used for the attainment of this purpose.

Keywords Human activity recognition · Machine learning · Fuzzy rule-based classifiers · Evolving systems · Wearable sensors · Business-oriented applications

1 Introduction

Ambient intelligence (AmI) applications ought to market brand new human centred applications. In addition, the increase of computational devices with powerful computing and communication capacities has made this offer much more realistic. Nowadays, we may talk about a realistic

pervasive or ubiquitous computing. It would be strange if some sort of intelligence would not have been involved in those technologies. Machine learning has brought to pervasive and ubiquitous application the step needed to create an ambient intelligent which is able to have some cognition of the environment and react accordingly to events that happen in it. As such, machine learning and Intelligent Systems are the technologies responsible to address these challenges. AmI should address several requirements such as personalization, adaptation, human-intelligible and transparency. In contrast to other recent works on the area; these requirements are taken into consideration in this paper by considering a novel way to use so-called evolving intelligent systems (EIS).

In this work, we have considered a very relevant application of AmI, namely human activity recognition (HAR). It is at the moment an intensive area of research. To accomplish this realistic AmI, it is not enough to know the characteristics of the environment but also properties of the subjects which are inside it at a certain moment of time. Consequently, the system has to be aware of their habits, independently of age, race or nationality and to be able to learn from them.

HAR systems have been recently developed using cameras and video signals (Chen et al. 2008). However these methods often presents underlying problems such as selecting a good viewpoint to track the person occlusion in the camera range, selecting a good view point to track the person, privacy connotations, etc. On the other hand, personal wearable sensor with high communication capacities such as mobile phones is a successful trend in the area. For example in Saponas et al. (2008) an “iPhone” is used to classify simplistic activities such as *running*, *walking*, *bicycling*. In other works such as (Tapia et al. 2007), more physical activities (including their intensities) are

J. Andreu (✉) · P. Angelov
School of Computing and Communications,
Lancaster University, Lancaster, UK
e-mail: j.andreu@lancaster.ac.uk

P. Angelov
e-mail: p.angelov@lancaster.ac.uk

recognized by means of several sensors attached to the human's body.

However, none of these previous works have considered the implication of these recognition systems in a general population market. Individuals can have different habits and same devices might be used by several individuals. In addition, new activities could appear. To cope with these problems an adaptive behaviour that produces an online updating of the learning model would be necessary. An evolving model has been adapted to deal with this problem. In a first attempt (Andreu et al. 2011), we tried to recognize simple activities using representative online features. In the present paper the pre-processing method has been improved, getting a better accuracy level and a bigger variety of recognized activities.

Evolving systems is a new machine learning concept which combines a self-adaptation of its structure with parameter learning. In this work has been used a fuzzy version of this type of model. It is named evolving fuzzy rule-based (FRB) type classifier (eClass) (Angelov and Buswell 2001) and (simpl_eClass) (Dutta-Baruah et al. 2011). To make use of this Evolving classifier, an advanced pre-processing mechanism based on a recursive version of principal component analysis (PCA)–linear discriminant analysis (LDA) has been developed specifically for HAR.

The remainder of this paper is organized as follows; in Sect. 2 is presented information about hardware and location of sensors; in Sect. 3 is given a detailed explanation about the appropriate pre-processing model; in Sect. 4 the evolving classifiers applied are described; Sect. 5 provides information for an unsupervised recognition of activities through model structure adaptation monitoring; Sect. 6 an evaluation of the approach's performance and finally a conclusion is provided in Sect. 7.

2 Hardware instrumentation

There are important advances in the miniaturization of electronics devices and in the amount of sensors; domestic devices such as smart phones or watches can carry on board. The accelerometer has become a very representative sensor which has made possible the appearance of multiple interactive phones applications. Accelerometers are classified by the normal of directions they are able to handle. Most current portable devices have embedded triaxial accelerometer. For such appreciation we have considered this type for this research. Data from the accelerometer is obtained in the following way: when a device is moving along an axis, the acceleration is positive; when the movement is towards positive coordinates, on the contrary, the acceleration is negative.

In a first study about the data in question we have considered two issues to take into account:

- (a) *Correlation of the data* accelerometer data presents highly correlation between axes. The influence of centripetal and gravitational forces leads to randomisation of the correlation either negatively or positively according to the movement.
- (b) *Sensitivity of the accelerometer* instinctive vibration, quick changes in the tilt and loose coupled location of devices, could lead to noise in the data sequence. Furthermore, it is not possible to recognize an activity precisely from a single data sample or a few samples. As a result of the high sensitivity of the sensor itself (for example 100 MHz). It is required to sensitive information for a period of time before recognition according to the position and intensity of the movement.

A suitable method to cope with these two issues will be explained in the following section.

Bandwidth of the accelerometer can be considered as another problem in order to get a good resolution of the movement. However, this problem can be easily solved using an appropriate set-up. In our case, best performance was achieved in a sample frequency of 150 MHz and accelerations between scales of accelerations over a scale of $\pm 6g$.

In particular, we considered an experiment with the following devices (Fig. 1):

SunSPOT this type of device is a miniature sensor with very good processing capacities, wireless communications and sensors on board which includes triaxial accelerometer accelerations over a scale of $\pm 6g$. This device is also programmable on board using (Oracle), Sun SPOT World (2011).

Nokia N97 a smart phone which can be used as a data collection platform. This device is very representative because it gives an idea of what a common personal device with elements of intelligence can achieve. It has on board a 3-axial accelerometer with handles a scale of $\pm 8g$. The device is fully programmable through Symbian OS (Wachenfeld et al. 2010).

2.1 Placement of the sensors on the body

Devices that a user can carry, such as phones, MP3 players, etc. can have different placements on the body. The location on the human body of the sensors that are part of these devices is very important in respect to facilitating the recognition of specific activities. For example, in case we aim to recognise whether someone is *standing up*, it is best to attach the sensor to the leg of the subject; however, in case we aim to recognise when an individual is "*writing on the white board*", a combination of sensors placed on the

Fig. 1 Devices prototypes used for evaluation and data collection



body's arms and legs would be required. Hence, the position of the sensor on the body is of crucial importance in order to collect informative signal for activity recognition using wearable sensors. The need to add multiple sensing points increases with the complexity of the activity.

It is clearly impossible to recognise human activity correctly based on a single sensor placed on one single part of the body. Much research has been carried out recently, aiming to clarify the number of sensors and locations where sensors can be applied to maximise the recognition rate. For example, statistical analysis has been used in Maurer et al. (2006) to discover the most significant placements of sensors while a heuristic method have been defined to cope with this problem. However, in real life applications, activity recognition should be made as pervasively as possible avoiding affecting individual's behaviours and habits. For example, it is expected not to force an individual to wear a sensitive object every time in the same place and same position in order to keep a certain level of freedom of movement for the user. Moreover, this individual cannot have the same motion habits/patterns as the other.

Consequently, the data that are collected from the sensor will not be the same for each individual. Therefore, personalisation is an important issue in HAR. Self-adaptive/evolving models (Angelov and Buswell 2001) can address this problem through self-learning classifiers aiming to recognise human activity of a specific individual, adapting to the personal specifics and self-developing their mechanism of classification. In evolving classifiers learning can take place on-line and the classifier structure evolves adapting to an individual's specifics.

3 Pre-processing and feature selection

3.1 Pre-processing

As a result of the chaotic nature of acceleration data, it is extremely complex to recognise human activities based on

the raw data. The main characteristic of a human activity is often hidden in the raw data. In addition, due to the non-stationary nature of the motion and other characteristics of human activity an adaptive approach (including in respect to feature selection) requires online update of classifier structure as well as of its parameters. This means the classifier structure (including its inputs–features) to be updated in parallel with the recognition task which means to evolve.

Pre-processing is an important stage of any machine learning and pattern recognition algorithm. It provides a valuable level of abstraction from raw data in respect to the classifier. This stage is usually done off-line on the basis of a large number of training data from a number of subjects (Bao and Intille 2004). This approach has obvious disadvantages in respect of the representativeness of the training data, the large amount of time and memory needed to process the data. It includes several tasks, such as; (1) outliers and missing data detection, (2) data cleaning and filtering, (3) data normalisation or standardisation, (4) feature extraction and feature selection. Pre-processing is often done off-line. It is important to mention that pre-processing can increase the recognition rate by up to 40%. In this study we propose and test several pre-processing methods which are particularly suitable for adaptive and evolving classifiers.

3.2 Features extraction

Feature selection aims to reduce the computation complexity and enhance recognition characteristics based on the analysis of the importance of each feature (usually in terms of statistical characteristics). Although real-time algorithms work in a sample-per-sample basis, for recognizing human activity from motion sensor it is recommendable to use short time windows as instances. For example, if the sampling frequency is 50 Hz a data instance represents 20 ms which is a very short period to get a representative motion pattern.

The features for the classifier that is supposed to recognise the human activity are usually different statistical characteristics of the signal. For example, Ravi et al. (2005) suggest using the following features:

- *Mean value of the signal* this represents the local posture of the body;
- *Standard deviation* it corresponds to the amount of motion presented in the signal. This is very valuable information to differentiate activities with a very different pattern such as *walking* and *running*, for example;
- *Energy of the signal* it represents the stress. The energy of the signal indicates the dynamics of the motion.
- *Correlation* it helps differentiate simple from complex movements. For example, both *walking* and *running* have similar acceleration pattern in all dimensions whereas *climbing stairs* has a very different pattern in two dimensions.

All these features are computed for a window with size of 256 samples with an overlap of 128 samples. This overlap was efficient to figure out repetitive cycles which can be significant in activities such as running, walking, vacuuming. Some works have also made use of this window and getting satisfactory results (Bao and Intille 2004; Ravi et al. 2005) (Fig. 2).

3.3 Online pre-processing

Due to the same physical principles, different accelerometer axes exhibit some correlation regardless of which human activity is being monitored. For example, significant values of the correlation ($R^2 = 0.4$) has been found between axes in human motion.

Principal component analysis it is one of the most known approaches for feature extraction and dimensionality reduction. It successfully addresses correlation data of raw input. It is obtained by means of orthogonal projections from a singular value decomposition (SVD) of the data matrix. By selecting a few components from these orthogonal projections, it is possible to transform the original data into a smaller set of uncorrelated values, maximizing its variance. After this transformation, obtained values are a linear transformation of the original values. Although less and uncorrelated, they preserve the majority of information about data variance contained into the original values.

PCA was widely applied in various areas, including spectroscopy, computing vision, intelligent sensors (Trevisan et al. 2010). The fundamentals of PCA can be summarized as follows: firstly, to get these projections a covariance matrix, C is computed and thereafter a new matrix, V of eigen-values and eigen-vectors, which diagonalise the previously computed matrix, are worked out.

Once the matrix V has been computed, eigenvectors, denoted as matrix w are sorted in decreasing order regarding their respective eigen-values. Data is standardized and projected onto the matrix, w .

Linear discriminant analysis as we stated above PCA is very useful in reducing the correlation and reducing the number of inputs to the classifier (thus the computational complexity). However, PCA does not take into account the labels or notations of the different activities, i.e. it takes only input data into account and not the class labels (outputs in terms of the classifier). In the case we want to apply some little pre-training or online user feedback, it will be necessary to make use of these labels. Once provided data is arranged based on the classes (activity types) that we expect to have so-called LDA approach can be applied. PCA rotates the inputs data in a way that maximises their separability and LDA helps finding these directions which will be best to distinguish between different classes (activities). The computation of LDA can be summarized as follow: the sample variance (σ^2) and sample variance per class (σ_c^2) are used to form scatter matrices (S_b and S_w). S_w denotes the within-class scatter matrix. Likewise, S_b represents a singular matrix denoted as between-classes scatter matrix. The separation between those matrices is defined as a measure of the signal-to-noise ratio for class labelling.

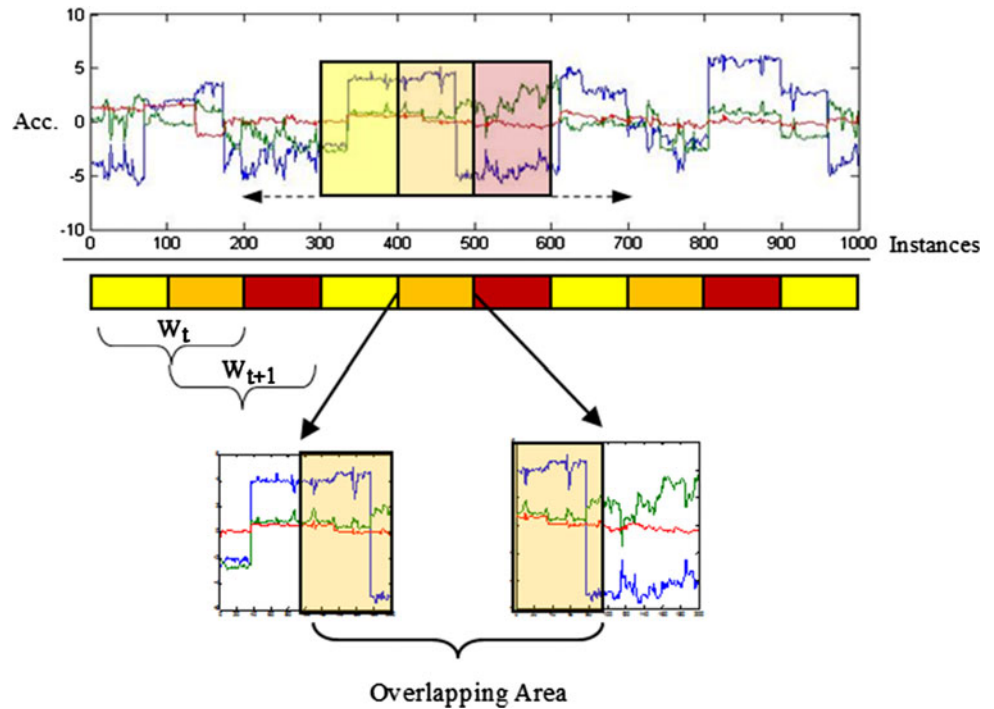
Hence, a criterion function J is computed on the basis of these two matrices, in the following form:

$$J(w) = \frac{w^T S_b w}{w^T S_w w} \quad (1)$$

Values of w which maximizes J are taken as desirable projections of data regarding not just their features (inputs to the classifier) but the class label (output) such as activity type.

On-line, recursive PCA, rPCA all methods described above (on its original versions) need to be computed off-line, they therefore required to be applied over some training data. This shortage will demand larger requirements in terms of space or computational cost. To cope with these issues, we suggest in this paper to use an online version of PCA-LDA, that is computed recursively (Dagher 2010). This techniques has been applied to face recognition problems, nevertheless it have not been used for a high noisy spectra such as acceleration or HAR systems. PCA and LDA are fully compatible methods because whereas PCA tries to find better projections of data according features, LDA finds what projections discriminate classes better. By difference from offline approaches of LDA, it is possible that when working online, labels could not be provided. Thus, we need to provide a number of expected classes or rules in the current instance of time. The algorithm can be summarized as follows (see Fig. 4):

Fig. 2 Sampling windows. Features presented in this point are not computed sample by sample (although possible). The best way to obtain a good description of the current motion and posture of the activity is by collecting samples in a moving window of around 3–5 s. The overlapping area gives the possibility of catching repetitive actions such as pedaling (cycling), swing arms (walking), folding legs (running), etc.



First PCA online PCA projections represented by the eigen-values, v can be calculated expressed as suggested by Dagher (2010) as follows:

$$v_k = \frac{1}{k} \sum_{i=1}^k u_i \cdot u_i^T \cdot \frac{v_{i-1}}{\|v_{i-1}\|} \quad (2)$$

where u denotes an orthogonal projection and k is the current time instant/sample.

This expression is based on the definition of v as $\lambda \cdot x = C \cdot x$ and $v = \lambda \cdot x$ where λ is the magnitude of the eigen-value, $\|v\|$ and C the covariance matrix.

This expression, (2) can be expressed recursively by weighting its current and last estimates by $\frac{k-1}{k}$ and $\frac{1}{k}$, respectively; then we get:

$$v_k = \frac{k-1}{k} \cdot v_{k-1} + \frac{1}{k} \cdot u_k \cdot u_k^T \cdot \frac{v_{k-1}}{v_{k-1}} \quad (3)$$

The algorithm estimates ortho-normalized vectors to ensure maximum independence of the directions. Thus, from the first order eigenvector a new one is computed by subtracting its orthogonal projection. The formula which describes this projection u_j is as follows:

$$u_k = u_{k-1} - u_{k-1}^T \cdot \frac{v_{k-1}}{\|v_{k-1}\|} \cdot \frac{v_k}{\|v_k\|} \quad (4)$$

Second On-line LDA, rLDA-PCA in order to get the desired projections in terms of LDA we have to calculate the between-, and within-classes scatter, S_b and S_w recursively:

$$S'_{wk} = S'_{wk-1} + (Y_k - (\sigma_c^2 - \sigma^2)) \cdot (Y_k - (\sigma_c^2 - \sigma^2))^T \quad (5)$$

$$S_{wk}^{-1} = \frac{S_{w(k-1)}^{-1} + (Y_k - (\sigma_c^2 - \sigma^2)) \cdot (Y_k - (\sigma_c^2 - \sigma^2))^T S_{w(k-1)}^{-1}}{1 + (Y_k - (\sigma_c^2 - \sigma^2))^T \cdot S_{w(k-1)}^{-1} \cdot (Y_k - (\sigma_c^2 - \sigma^2))} \quad (6)$$

$$S_{wk} = A_{k-1} \cdot S'_{wk} \cdot A_{k-1}^T \quad (7)$$

$$S_{wk}^{-1} = A_{k-1} \cdot S_{wk}^{-1} \cdot A_{k-1}^T \quad (8)$$

$$S_b = A_{k-1} \cdot \left(\sum_{i=1}^c n_i \cdot (\sigma_c^2 - \sigma^2) \cdot (\sigma_c^2 - \sigma^2)^T \right) \cdot A_{k-1}^T \quad (9)$$

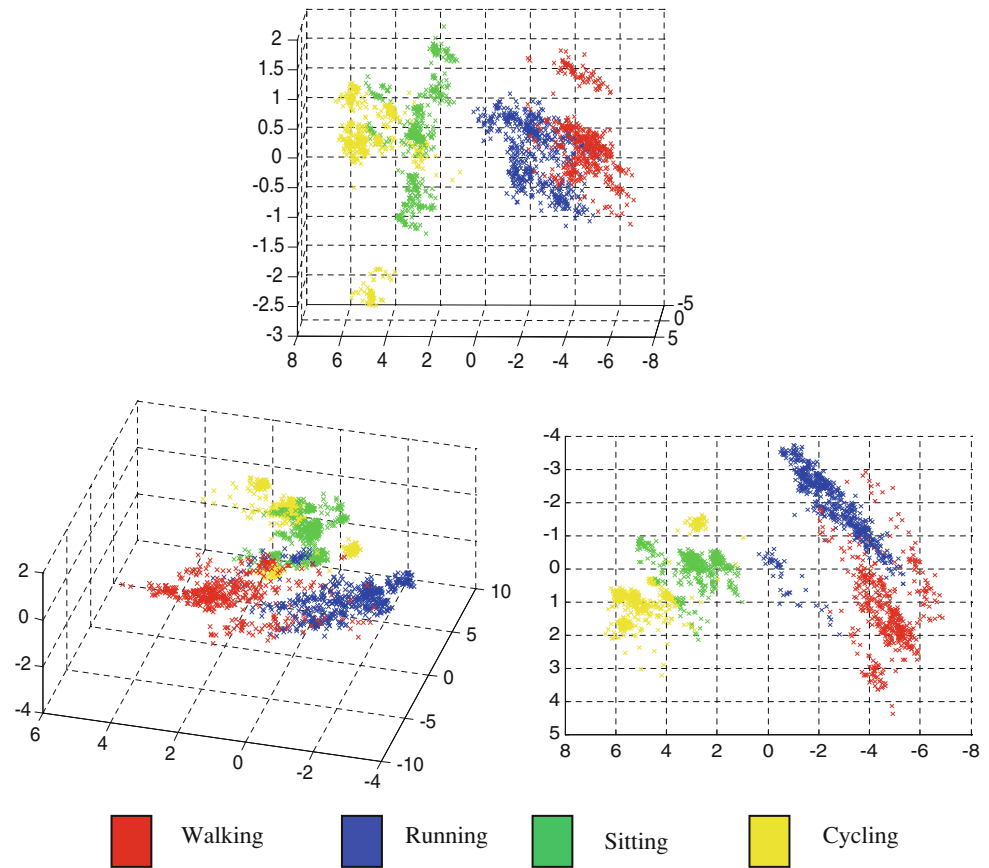
where $Y_k = x_k - \mu$ is the variance; A_k is the matrix of desired orthogonal projections (eigenvectors) S_w^{-1} and S_b .

The joint of these two calculations (orthonormal projections and then scatter matrices) will make up the rLDA-PCA algorithm as such. An explanatory diagram of the combination of both methods in a same time step, k is shown in Fig. 4. At the end of the algorithm desired eigenvectors which guarantee best projections and features that enhance the separation between classes of activities are provided. Consequently, the raw data, x are being mapped onto the feature space, f (see Fig. 3).

4 An evolving classifier for HAR

In this paper we use evolving FRB classifiers of the eClass family (Angelov et al. 2007), namely *eClass1* (Angelov

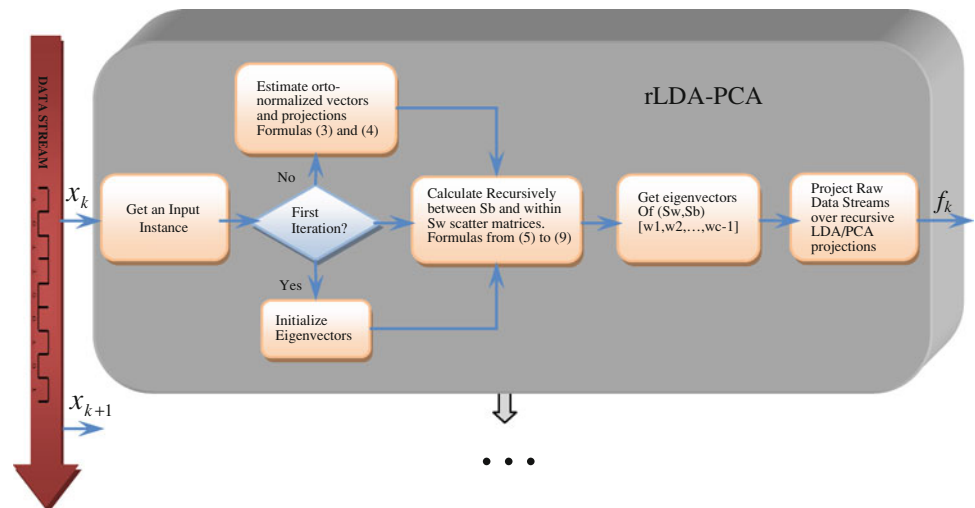
Fig. 3 Features grouped by labels after rLDA-PCA was applied. As you can see in the different plots it is possible to get well-separated prototypes of after applied a method that maximizes the variance between them. This is because an rLDA-PCA method exploits the difference of intensities between activities that can be represented by the standard deviation and energy of the acceleration



and Zhou 2008), and a simplified version called *simpl_eClass1* (Dutta-Baruah et al. 2011). Normally, classifiers use evolutionary algorithms or gradient-based schemes to be trained off-line. Some examples of this are NN combined with back propagation. EIS such as the ones proposed here, are designed to be fully on-line in estimating and training. Both proposed classifiers have self-learning and self-developing capacities, adapting not just

its parameters but the whole structure of their model. This is achieved by means of an evolving (self-developing) FRB structure. This FRB structure is created from data-stream inputs $z = [f^T, L]^T$. The model structure is FRB of Takagi-Sugeno type. The output of the model can be considered as a nonlinear estimation. It might be displayed as multi-input multi-output (MIMO) model with rules of the following form:

Fig. 4 rLDA-PCA scheme. Data inputs X_k are obtained from the data stream and pre-process in the rLDA-PCA module. Reduced mapped data from computed component will be forwarded to the evolving learning approach. The rLDA-PCA module only retains information from S_b and S_w and projections matrix W but raw data samples from accelerometers sensors are not stored in the memory. The time of computing the rLDA-PCA algorithm is around 0.0017 s



$$\text{Rule}^i : \text{IF}(x_1 \text{ is } \sim x_1^i) \text{AND} \dots (x_n \text{ is } \sim x_n^i) \text{THEN}(\mathbf{y}^i = \mathbf{x}_e \theta^i) \quad (10)$$

where $\mathbf{y}^i = [y_1^i \ y_2^i \ \dots \ y_m^i]$ is the m -dimensional output of i th rule, $\mathbf{x}_e = [1 \ x_1 \ \dots \ x_n]$ is the extended input vector, θ^i is the matrix of local sub-system parameters.

This set of fuzzy rules (note that, due to its evolving nature, their number N is not fixed) defines with its antecedent/premise (IF) part a 3-D feature space, $f \in R^3$ (from three LDA-PCA components) and with its consequent part it determines in a fuzzy way (with a degree of fulfilment) the class label (activity), $L_i, i = [1, K]$.

Thanks to this structure we can consider that evolving fuzzy classifiers applied to HAR provide the following benefits:

1. Starting from scratch, the fuzzy rule-base structure self-develops itself. On the other hand, the structure of a conventional classifier is determined off-line (and then fixed).
2. Additionally the on-line learning process considers this flexible rule-base structure.

4.1 Learning eClass

Focal points are generated to describe the nature and dynamics of the activity signal. This process of self-calibration resembles adaptive control and estimation of the sensors. Two phases are formed during the process:

1. Activity estimation (classification);
2. Update or evolution of the model.

In the first phase the class label is unknown, consequently it is estimated. During the second phase this same class label is used as a supervisory information to update the model structure (so-called process of structure evolution), as well as its internal parameters.

The overall fuzzy rule-base structure has a K number of sub-rule based with the same number of consequents. However, the number of rules must be not less than the number of activities ($N \geq K$). That is to say, every data from a non-previously seen label, automatically becomes a prototype. Model updating procedures (replacement and removal of rules) makes this situation temporal. Later, this prototype is usually replaced with more descriptive prototypes. In this sense, *eClass* learns from the scratch and a fixed number of activities do not need to be set during the whole life cycle of the application. Consequently, an activity does not have a framed FRB scheme to be described. Activities can be described by several prototypes that represent a particular action, pose or movement of a particular activity. The partitioning model is generated by a density model

that defines the potential or density of each point in the feature space.

The use of a density model comes from the following principle: “every point with the highest density is chosen to be the focal point (prototype) for the antecedent of a fuzzy rule”. By means of this procedure, it is possible to define rules with high descriptive power of the activity.

The data density model needs to be updated each time a new data sample is added; logically, density values of the existing prototypes have to be updated. This procedure is also done in a recursive way.

The rules of the classifiers are formed around focal points (prototypes centers), where a focal point is a representative of all the data samples within a cluster. The consequent parameters of the rules are estimated using weighted recursive least squares (wRLS) approach (Angelov and Buswell 2001).

4.2 Learning Simpl_eClass

Simpl_eClass1 accomplishes the same process and characteristics described in the previous point. It differs from *eClass1* mainly due to the learning process, where it is not needed to build a density model (Dutta-Baruah et al. 2011). *eClass* needs to compute the potentials of each new acceleration sample that comes into the system plus update the potentials of all the previous prototypes. The computational complexity of *eClass1* is $O(N)$, where N is the number of activity prototypes (respectively, fuzzy rules). On the other hand, to determine a prototype in *simpl_eClass1*, it is only required a mean value calculation. It can be computed online as each sample arrives. Hence, in *simpl_eClass1* the computational complexity is reduced from $O(N)$ to $O(1)$ —by a factor of N . A detailed description of *simpl_eClass1* is presented in Dutta-Baruah et al. (2011). Nevertheless, description of prototypes is reduced to a mean value. The computational complexity is reduced slightly but the total classification of activities can also be penalized. We propose this approach for devices with very small processing capacities. This approach is specially defined for wearable computing where very small Nano circuits that can be embedded in buttons or small garments.

5 Unsupervised discovery of activities through evolving model monitoring

The evolving model can be evaluated using a representative pre-training from a cross-sectional analysis of the data. This data does not have to be fully representative (expert knowledge), this means that the model can be pre-trained using just information from one source and applied to others as a result of its evolving behaviour. On the other

hand, another method is based on clustering, that is to say evolving from scratch and also estimating activities at the same time.

In HAR context, it might be presented the following case study: a wearable device is programmed with an EIS on board and trained in the factory with a sample dataset; next, it is out to be sold to the public. In another context the device could be simply manufactured and programmed but not trained and just sold out to users. If we consider this situation, the EIS updates itself to the activity patterns of its users but without any prior knowledge involved.

5.1 Detect drift and shift in the data streams

As it has been explained in the previous section, the evolving model creates a FRB model from data streams inputs. Focal points are dynamic and they are able to embody a particular movement of an activity. Some activities demand the combination of several sub-task movements which can constitute the standard pattern or shape of the activity. In Andreu et al. (2011) patterns of activities by evolving clustering techniques were studied. This patterns form a well-defined descriptor of each one of the activities. Users can perform sequential activities. It is complex to determine exactly when an activity has turn into other. It is usually a step-by-step process or a combination of different activities. In Lughofer and Angelov (2011), similar techniques are used to detecting drift and shift to detect errors automatically. Defining this previously explained pattern as standard descriptor of the activity plus this automatic detection of anomalies, it is possible to detect that there is an alteration in the standard shape it is being formed in a particular instance of time. For this purpose, age and support of clusters which conforms the shape are monitored. These *age* and *support* measurements can be used to analyse the quality of clusters and their respective fuzzy rules. Support can be determined by a simple counting of the associated samples to a prototype (nearest points to prototype). Its formula is as follows:

Each time a new rule appears, its age is initiated by the current instance of time k . When a new point is clustered into an existing prototype (distance from a point to this prototype is the smaller between all the other clusters), their age gets smaller. If there is no sample assigned to those prototypes, they will become older rules by one. The age of a fuzzy rule can take values between a range of $[0; k]$. In Fig. 5 the age of 11 prototypes are displayed. The changes of this aging rate can automatically detect such changes and respond by adapting the model. It is easy to see precisely when *shift* happens. A *shift*, in the density space model, describes a change on the density in a determined region of the space. This change of density in

the density space can be used to describe online when a change in the pattern shape has happened. This change might be due to a change in the movements of the activity. After analysing data shift and swap between activities for a possible synchronization, we discovered than when half of the prototypes experience a shift, it is highly possible that a data shift occurs.

Another option is to study which prototypes experience drop in its age when an activity happens. These prototypes are the most representative for that activity in particular. Consequently, considering that those prototypes (or in fuzzy also called focal point of respective fuzzy rules, see Fig. 7) the ones with a higher probability of activation for an activity. In Fig. 6 it is displayed a histogram of activities regarding that clusters have a higher level of activation for each of these four activities (running, walking, cycling and sitting). Note that there are some activities that clearly cause a *shift* in determined clusters. This is the case of cluster 5 which is used to gather lots of points into the region covers, when a new activity occurs. In addition, as a result of the evolving behaviour across the life of system, an activity becomes complex in the sense that new typical types of movements and positions which belong to the same activity are recorded.

At the time of the online activity recognition system is running, the concurrent shape of cluster might not match fully any of the previously show patterns of activities. This fact is because the online shape could be a mixture between all activities that have happened in the past.

6 Results evaluation

In this section the proposed classification scheme for HAR which includes recursive PCA and LDA as well as the evolving FRB Classifier (*eHAR*) has been evaluated. *eHAR* includes two learning models *eClass* and *simpl_eClass*. This model is compared with several well-known classifiers. These classifiers are organised in two groups: (1) static, off-line classifiers, and (2) on-line and evolving classifiers. The former, generally, provide higher classification rate, but impose heavy requirements on the memory and computational resources. They keep most of the data in the memory and process all of them at once. On the other hand, the latter process the data sample by sample (on-line) and may be able to adapt their structure (evolve) as a reaction to the variance in the data pattern. As a result of this adaptation and evolution they result in less complex models and due to the on-line processing manner they require significantly fewer computational resources. For this reason, they use to have a better performance in terms of CPU and memory usage which compensates the somewhat lower accuracy.

Fig. 5 Rule’s age evolution. Appearance of new rules and shift of existing rules (model spots) are clear signs of a change in the context of activities. Monitoring the age is possible to detect automatically that an “event” has happened. This indicator may be useful to request external class feedback from other sensors or discard the last recognized activity. Further information of these tendency shifts can be done by analysing the current gradient of ages per prototypes. Usually, shifts that compromise a change in the context display a “smooth drift” and parameter tuning “sharp shift”

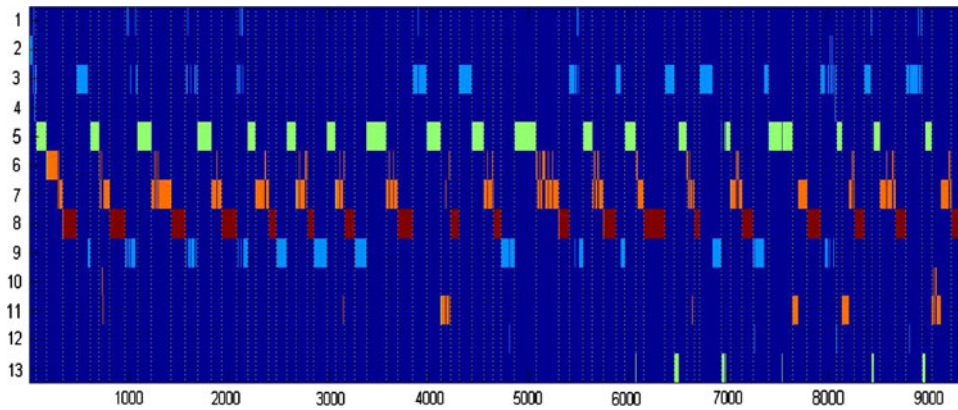
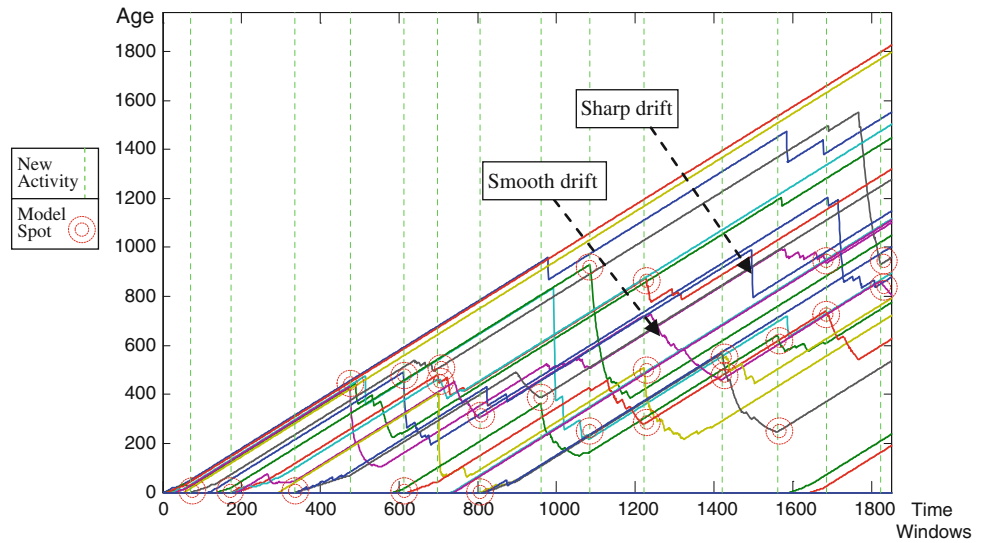
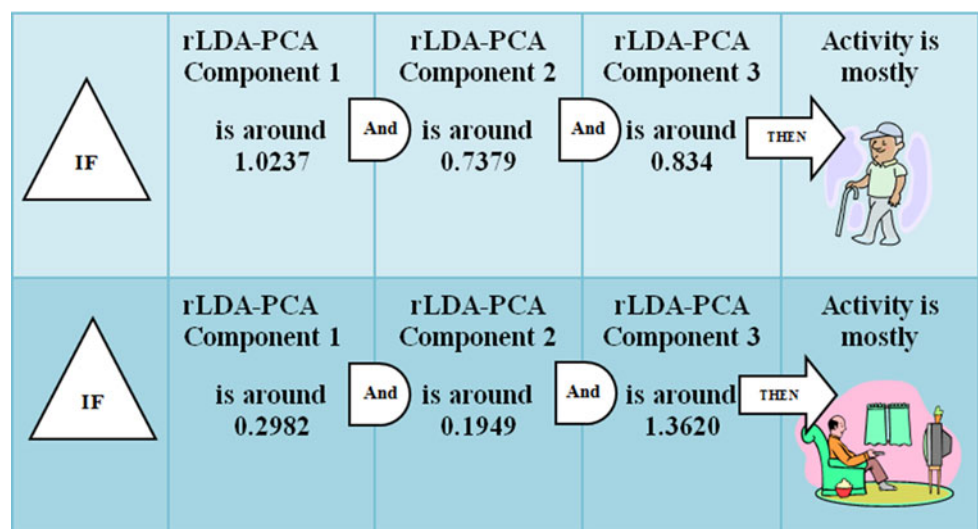


Fig. 6 Rule’s support histogram grouped by activities. Each prototype is assigned to be mostly representative for an activity. That is to say that points which belongs to an activity are most likely to be included inside these prototypes. Activities can be represented by

several prototypes. For example prototypes 6–7 and 2–3. This is because an activity can be composed of several sub-actions or sub-activities and intensity also varies

Fig. 7 Rules can be representative in a human understandable way, following the linguistics (IF-AND_THEN) represented in this figure. The three coefficients are resultant of mapping the feature values (over a sliding window between 3 or 5 s) into the rLDA-PCA space components. Thereafter rule’s antecedents are aggregated in order to know what the most representative activity (output) is



All the results of the comparison are tabulated in Table 1. As it can be seen, the proposed *eHAR* classifier is very competitive. It has very low memory requirements and CPU usage. Classification rate is also higher than that of any other on-line classifier and very competitive with the off-line classifiers.

In Fig. 8, a confusion matrix is presented which includes all activities that have been considered. It can be seen that the general activities such as walking, running, lying down or even sporty ones (such as bicycling) are classified with higher rates. On the other hand, there are also two intermediate activities that are composed of several sub-activities. These are vacuuming and stretching. The classification rates for these activities are fluctuating because they involve several sub-activities. In order to overcome this issue, larger windows or online feedback from objects via integrated sensors would help to evolve the model towards a satisfactory level of interpretation of these activities.

6.1 Improving recognition rates through occasional real-time user feedback

Figure 9 shows the evolution of the recognition accuracy levels. At the beginning blue and green lines experiment a high drop in the accuracy levels. In this part of the process we are getting data from the first subject of test. We may say the model is building itself from scratch. It is evolving and self-learning an appropriated model structure to become general to all the other subjects. The model starts generating rules until a total of 21. During the first part of the testing session, the model generates a total of 15 rules but in the course of test, as long as the experiment is digesting data from different new users, only 6 new rules are generated. This online forming of rules makes the

classifier to self-tune and self-enable its parameter. In order to control the maximum volume of the model itself, it is possible to remove rules using the same age monitoring of rules we explained in the previous sections. Rules with a very old age will be removed with the purpose of fostering the younger ones. As a result of the online/real-time capacities of the system, it is possible to enable online training on-the-fly along the course of the running process. This little update periods (a few seconds) can boost the recognition close to a 10% improvement.

As it has been said about Fig. 9, the recognition rate drops during the first subject of test (the classifier can start with a training period on a different set of subjects or ‘from scratch’). In this experiment, a training period was not applied. Following the first subject, the classifier self-tunes its parameters as well as self-learns its structure (evolves) and, as a result, raises its classification rate. After an increase to values around 70–80% there is some saturation (and even reverse). This is due to the known phenomena in signal and data processing that ‘more data not always brings more useful information’.

The blue line in Fig. 9 illustrates the effect of self-learning by temporal re-training (user or external sensors feedback). The algorithm is able to work online and with very little computational cost. Thanks to those abilities it is possible to provide in real-time ground truth (activity labels). Several such update periods (with duration of 1–2 min) were used in real-time. It is obvious that they helped *eHAR* increase the classification rate to 83–84%, which surpasses all off-line classifiers (with a fixed model built up during its training period). It is due to the fact that the classifier adapts its structure (fuzzy rules) to better match the nature and dynamic of the coming data stream from wearable sensors.

Table 1 Comparison table

	Memory (KB)	Max. CPU (%)	Class. rate (%)
Off line classifiers			
Hoeffding adaptive tree	2.3	10	81
Incremental Naïve Bayes	4.868	14	76.53
Nearest neighbour (euclidean distance)	3.64	47	72
Kstar (entropic distance measure)	4.624	50	83.55
Locally weighted learning + Naïve Bayes	1.853	48	75.16
On-line classifiers			
OzaBagASHT (adaptive tree)	0.105	18	70.53
OzaBagADWIN (adaptive windowing + adaptive tree)	0.992	13	70.61
SingleClassifierDrift (drift detection + adaptive tree)	0.012	8	49.99
SingleClassifierDrift (drift detection + incremental Naïve Bayes)	0.01	6	63.41
AdaHoeffdingOptionTree (adaptive tree + incremental Naïve Bayes)	0.01	8	70.45
eHAR (proposed) (fuzzy type classifier)	0.02	5	71.4
Simpl_eHAR (proposed) (fuzzy type classifier)	0.01	4	70.2

Proposed methods are highlighted with bold



Fig. 8 Confusion matrix of recognized activities

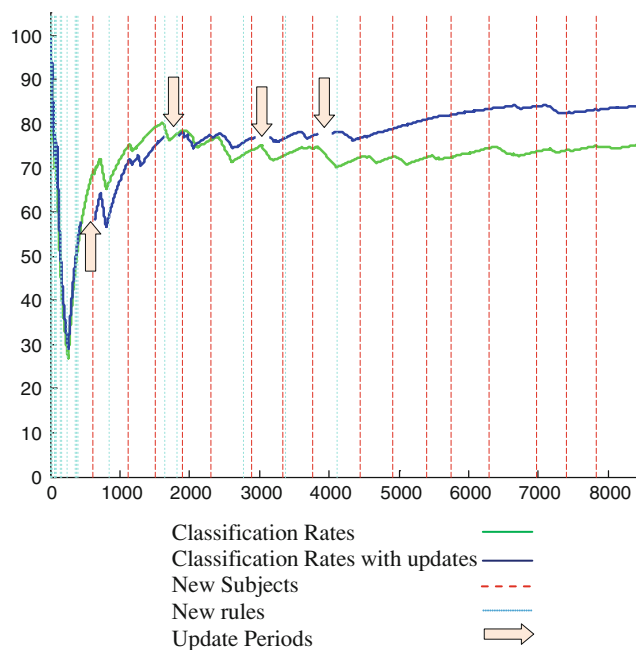


Fig. 9 Evolution of the classification rate evolution (the effect of update by eHAR is clearly seen to improve the classification rate despite switching to new subjects)

7 Conclusion

In this paper have been described the groups of a new approach of processing HAR (eHAR). It takes into account the heterogeneity of the population by using an evolving model which is able to adapt itself to any circumstance. People have different complexities and habits, devices are personal but cannot be distributed using a standard model which represents that heterogeneity. We believe EIS as are

the right way to achieve a satisfactory and marketable product. On the other hand, evaluation and results have showed that there is a minimal difference in terms of accuracy between other methods used such as (incremental Naïve Bayes, adaptive decision trees, updateable Bayesian networks...) and evolving fuzzy system. However, there is also an important gain in terms of computational resources needed to run the whole recognition system and much more gain compared with offline approaches. This advantage makes this system ready to be embedded in very low-capacity devices such as miniaturised buttons or garments. In addition, as a result of the fuzzy type classifiers applied, the information obtained is human-intelligible (linguistic, see Fig. 7) and the fundamentals of a very convenient machine learning approach to be added to an evolving (population concerned) HAR architecture.

References

- Andreu J, Dutta Baruah R, Angelov PP (2011) Real time recognition of human activities from wearable sensors by evolving classifiers. In: IEEE international conference in fuzzy systems (FUZZ), pp 2779–2785. doi:[10.1109/FUZZY.2011.6007595](https://doi.org/10.1109/FUZZY.2011.6007595)
- Angelov P, Buswell R (2001) Evolving rule-based models: a tool for intelligent adaptation. In: Proceedings of joint 9th IFSA world congress and 20th NAFIPS international conference, vol 1–5, pp 1062–1067. doi:[10.1109/NAFIPS.2001.944752](https://doi.org/10.1109/NAFIPS.2001.944752)
- Angelov PP, Zhou X (2008) Evolving fuzzy-rule-based classifiers from data streams. IEEE Trans Fuzzy Syst 16(6):1462–1475. doi:[10.1109/Tfuzz.2008.925904](https://doi.org/10.1109/Tfuzz.2008.925904)
- Angelov P, Zhou XW, Klawonn F (2007) Evolving fuzzy rule-based classifiers. In: IEEE symposium on computational intelligence in image and signal processing, pp 220–225. doi:[10.1109/CIISP.2007.369172](https://doi.org/10.1109/CIISP.2007.369172)
- Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. Proc Pervasive Comput 3001:1–17. doi:[10.1007/b96922](https://doi.org/10.1007/b96922)
- Chen W, Aghajan H, Kleihorst R (2008) Real-time human posture reconstruction in wireless smart camera networks. In: International conference on information processing in sensor network, pp 321–331. doi:[10.1109/IPSAN.2008.20](https://doi.org/10.1109/IPSAN.2008.20)
- Dagher I (2010) Incremental PCA-LDA algorithm. In: International conference on computational intelligence for measurement systems and applications, pp 97–101. doi:[10.1109/CIMSA.2010.5611752](https://doi.org/10.1109/CIMSA.2010.5611752)
- Dutta-Baruah R, Angelov P, Andreu J (2011) Simpl_eClass: simplified potential-free evolving fuzzy rule-based classifiers. In: IEEE international conference on systems, man, and cybernetics (in press)
- Lughofer E, Angelov P (2011) Handling drifts and shifts in on-line data streams with evolving fuzzy systems. Appl Soft Comput 11:2057–2068. doi:[10.1016/j.asoc.2010.07.003](https://doi.org/10.1016/j.asoc.2010.07.003)
- Maurer U, Smailagic A, Siewiorek DP, Deisher M (2006) Activity recognition and monitoring using multiple sensors on different body positions. In: International workshop on wearable and implantable body sensor networks, pp 4–116. doi:[10.1109/BSN.2006.6](https://doi.org/10.1109/BSN.2006.6)
- Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. In: Proceedings of the 17th

- conference on innovative applications of artificial intelligence, pp 1541–1546
- Saponas S, Lester J, Froehlich J, Fogarty J, Landay J (2008) iLearn on the iPhone: real-time human activity classification on commodity mobile phones. Technical Report. University of Washington
- Sun SPOT World (2011) Oracle Labs <http://www.sunspotworld.com/>. Accessed 20 July 2011
- Tapia EM, Intille SS, Haskell W, Larson K, Wright J, King A, Friedman R (2007) Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: 11th IEEE international symposium on wearable computers, pp 1–4. doi:[10.1109/ISWC.2007.4373774](https://doi.org/10.1109/ISWC.2007.4373774)
- Trevisan J, Angelov PP, Carmichael PL, Scott AD, Martin FL (2010) A mathematical framework for spectroscopy data analysis to characterize chemical-induced alterations in the (SHE) assay (pH 6.7). *Analyst* 135:3266–3272. doi:[10.1039/C0AN00586J](https://doi.org/10.1039/C0AN00586J)
- Wachenfeld S, Madeja M, Jiang XY (2010) Developing mobile multimedia applications on Symbian OS devices. In: *Lecture notes in computer science*, pp 238–263. doi:[10.1007/978-3-642-12349-8_14](https://doi.org/10.1007/978-3-642-12349-8_14)