Light-Weight Online Unsupervised Posture Detection by Smartphone Accelerometer

Özgür Yürür, *Member, IEEE*, Chi Harold Liu, *Member, IEEE*, and Wilfrido Moreno

***Abstract*—This paper proposes a light-weight online classification method to detect smarthpone user's postural actions, such as sitting, standing, walking, and running. These actions are named as "user states" since they are inferred after the analysis of data acquired from the smartphones equipped accelerometer sensors. To differentiate one user state from another, many studies can be found in the literature. However, this study differs from all others by offering a computational lightweight and online classification method without knowing any *priori* information. Moreover, the proposed method not only provides a standalone solution in differentiation of user states, but also it assists other widely used offline supervised classification methods by automatically generating training data classes and/or input system matrices. Furthermore, we improve these existing methods for the purpose of online processing by reducing the required computational burden. Extensive experimental results show that the proposed method makes a solid differentiation in user states even when the sensor is being operated under slower sampling frequencies.**

***Index Terms*—Mobile sensing, posture detection, unsupervised learning.**

I. INTRODUCTION

THE understanding of human activity is based on the discovery of the activity pattern and accurate recognition of the activity itself. Therefore, researchers have focused on implementing pervasive computing systems to infer activities from unknown activity patterns, which are defined as the extracted context from smartphone-equipped sensors. The existence and awareness of the context provide the capability of being conscious of the physical environment or situation around device users. Such awareness also makes network services respond proactively and intelligently. This leads to the exciting vision of forming smart spaces, i.e., the Internet of Things [1], [2], which allows applications to encourage users to collect, analyze, and share numerous local sensory knowledge for a large-scale community use, by creating a knowledge network that is capable of making autonomous logical decisions to actuate environmental objects.

Especially, the ever-increasing technical advances in embedded systems along with proliferation of growing small-sized sensor development and deployment have enabled smartphones to be re-purposed, in order to recognize the daily occurring human-based actions, activities, and interactions. It is believed that recognizing human-related event patterns, called *user states* in this paper, accurately enough could give a better understanding of human behaviors. Therefore, recognizing human-centric activities and behaviors have been an important topic in pervasive mobile computing. Human activity recognition (HAR) [3], [4] intends to observe human-related actions in order to obtain an understanding of what type of activities/routines that individuals perform within a time interval. Providing accurate information about HAR-relevant data history could assist individuals to enhance the quality of life such as having better well-being, fitness level, and situational-awareness [5]–[7]. For example, patients with diabetes, obesity, or heart disease are suggested to follow a predefined fitness program as part of their treatment [8]. In this case, information of human postures and movements can be inferred by an HAR system to provide useful feedbacks to the caregiver from behavior analysis.

Manuscript received December 12, 2014; accepted February 15, 2015. Date of publication February 20, 2015; date of current version July 30, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61300179. *(Corresponding author: Chi Harold Liu.)*

O. Yürür and W. Moreno are with the Department of Electrical Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: oyurur@mail.usf.edu; wmoreno@usf.edu).

C. H. Liu is with the School of Software, Beijing Institute of Technology, Beijing 100081, China (e-mail: chiliu@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2015.2404929

II. RELATED WORK

Many studies have been conducted to detect user-centric postural actions within the concept of HAR [9]–[16] by using accelerometers, Wi-Fi, GPS, or other smartphone-equipped sensors. In this paper, we only study the smartphone accelerometer in an HAR-based analysis, which makes the analysis more challenging due to the lack of data fusion of multiple sensors, and also due to difficulty in data calibration rooted from signal distortion, caused by frequent changes in smartphone orientations.

A statistical tool-based classification, mostly using hidden Markov models (HMMs) [17]–[19] or using autoregressive (AR) models [20], is one of the foremost methods to detect user-related physical activities. It exploits the context obtained from wearable or built-in mobile device sensors. However, these studies mostly apply predefined or user-manipulated system parameter settings, such as pretrained state transition matrix in HMMs or filtering coefficients in ARs. They are not suitable for online processing, due to the increasing computational workload while enlarging data size. Also, fixed system parameters do not provide robustness to adapt the time-variant nature of user activity behaviors.

On the other hand, other studies rely on creating feature vectors as the first step to exploit signal characteristics of sensory data. A feature vector consists of many signal processing functions starting from mean, standard deviation, correlation, to frequency, and wavelet transform models [21]–[24]. Then, they attempt to classify feature vectors according to the predefined specific data classes. Accordingly, after creating a high-dimensional feature vector, pattern recognition algorithms are applied to find out the hidden context inside the feature vector. The major drawback of these algorithms stems from an *offline* decision process, as follows. First, all sensory observations are recorded. Second, feature vectors are constructed by partitioning data records with a predefined window length. Third, the selection of default feature vector for specifying a *training* data class is *user-manipulated*, which means the feature vector is extracted from a specific (e.g., visually observed) window of fully recorded data. Finally, as called *testing*, is to use a template matching algorithm by mapping instantly constructed feature vectors into default feature vectors. Pattern recognition techniques for clustering diverse data classes, such as k-means [25], k-nearest neighbors (k-NN) search [26], and support vector machines (SVMs) [20], are involved in this final step to infer the context. Unfortunately, the given clustering techniques are not efficient while processing large data clusters. Also, SVMs cannot deal with multiclass classification directly. The multiclass classification problem is usually solved by problem decomposition into several two-class problems [27]. Furthermore, pattern recognition toolkits such as WEKA[1] are also used to obtain the offline running classification results [28], [29].

Toward this end, this paper proposes an online solution that fully exploits acceleration signals within the fast decision tree (DT) classifiers, without setting any predefined/fixed thresholds over any specific acceleration spaces, in order to differentiate user activities. DT-based classification is used in almost every other studies, whereas our proposed classification method provides the following novel properties, which also make findings in this paper differ from other studies such as [12]–[14], [20], [23], [27] under a similar name:

1) *unsupervised learning*: no *priori* information, no fixed thresholds, no initial training data classes;
2) *adaptive*: robust solution to a changing orientation of the device;
3) *light-weight*: efficient tree-based classification by applying sufficient signal processing techniques: no redundant computational workload;
4) *online*: instant context inference;
5) *assisting*: working standalone and/or assisting other classification algorithms by creating training data classes or input matrices;
6) *updating*: computationally efficient update/add/delete process on training data classes.

Our proposed solution also enhances some widely used supervised classification methods, such as Gaussian mixture models (GMMs), k-NN, linear discriminant analysis (LDA), for online processing by providing training data classes *without a prior offline process* as well as *supporting the observation analysis* defined in statistical-based tools such as HMMs. The reason is fourfold. 1) The construction of a feature vector generally employs a huge number of signal processing primitives (e.g., usually including around 40 features), and these features are mostly randomly chosen without analyzing processed signal characteristics in detail. On the contrary, our solution only offers 15 features. 2) We do not use an offline process to specify which signal time frame belongs to which activity. Our solution recognizes activity and creates related training data classes on-the-fly. 3) We set the transition probability matrix in HMM analysis with equal probabilities among state transitions. However, observations collected by our solution route the evolution of HMM chain. 4) There are two important issues in mobile online processing. One is to have less computational complexity for energy efficiency, and the other is being adaptive to heterogeneous sensory readings. Therefore, whenever our solution predicts a high accuracy in activity recognition, it updates training data classes, that allows to collect dynamic information of activity patterns. Also, we use computationally efficient algorithms to update existing training data classes instead of regenerating them over and over.

III. Proposed Classification Method

The complete system architecture is given in Fig. 1. Accordingly, a sequence of sensory data is collected by a sliding window. To be able to infer the hidden context (i.e., the user states, as sitting, standing, walking, and running), there are two suggested modes: 1) standalone or 2) assisting mode.

Standalone mode uses a novel classification algorithm proposed in this paper. It provides a lightweight, online, and unsupervised context inference solution by using a sufficient number of statistical/signal processing techniques. It is not only capable of providing self-decision mechanism on context inference for mobile device user postures, but also helps other mostly used context recognition/classification methods, by preparing them for unsupervised learning and improving them for online processing. Therefore, our solution produces training data classes or system matrices to represent the context and decides feature selections to be used by these classification methods.

Assisting mode is designed for any other studies that intend to integrate our proposed method into their own works. Most studies employ context inference methods that are based on widely approved pattern recognition techniques or statistical tools. However, these methods are generally lack of offline processing for resource-constrained mobile platforms. Therefore, we propose the assisting mode. It receives sensory data sequences as input along with having *a prior information* provided by standalone mode. The inputs undergo a feature extraction process, whose functions are already defined by the standalone mode, to build a corresponding feature vector. Note that additional features could be added to this process to have more accurate results, but we only use what standalone mode produces at this point. Then, context inference is made in the assisting mode, by using diverse classification methods such as GMM, k-NN, LDA, or HMM. In addition, another powerful property provided in this mode is to be able to update data sets dynamically in a computationally efficient way for online processing. The update occurs whenever the standalone mode

[1][Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

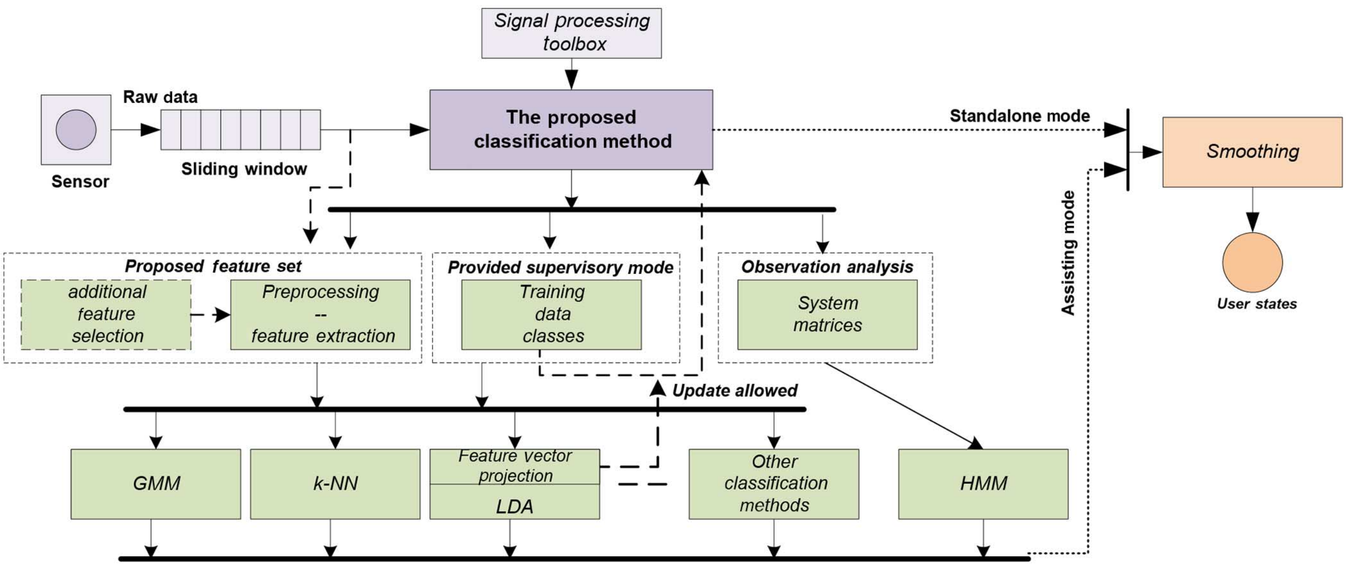

Fig. 1. Proposed structure for lightweight, online, and unsupervised user state classification: standalone or assisting modes.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS USED IN THIS PAPER

Symbol	Definition (section where the symbol is first used)
$\{x, y, z\}$	Accelerometer sensor readings (i.e., three-axial info) (IV)
i, j	Indexes for the accelerometer axises (IV)
u, t	Time indexes (IV)
L	Length of sliding window (IV)
f	Active sliding window at current time (IV)
f^{P}	Previously active sliding window (i.e. $L/2$ samples earlier) (IV)
x	Feature vector (V-A)
$s, s^*, \acute{s}$	Indexes for user state classes (V-B1)
n, m	Indexes for a data point in a class/feature vector (V-B1)
W	Feature projection matrix (V-B3)
y	Feature projection vector (V-B2)
μ, m	Mean values (IV)
σ	Standard deviation (IV)
Λ, S	Covariance matrices (V-B3)
A, B	Between and within class scatter matrices (V-B3)
a, b	User state transition and observation matrices (V-C)
$n, \acute{n}$	The number of samples (V-B3)

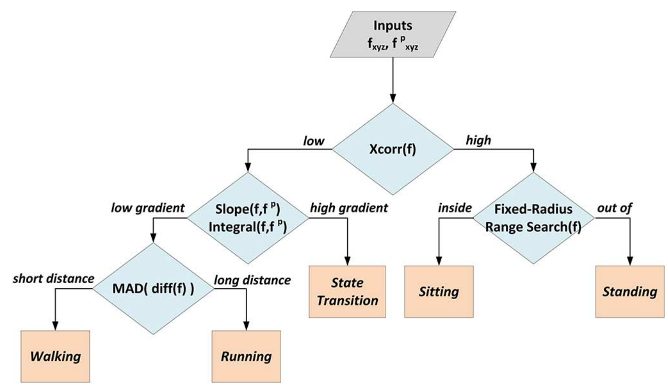

Fig. 2. Proposed DT-based classification method: standalone mode.

makes a solid differentiation in user states. Therefore, adaptability can be achieved toward changing user behavior profiles or time-variant sensor signal characteristics.

This paper is organized as follows. Section IV describes a novel DT-based classification method designed for posture detection, which is used in standalone mode. The introduction of widely used classification methods and their integrations with the proposed classification algorithm, which is used in the assisting mode, are explained in Section V. This section also provides the updating procedures in online processing, to help each classification method achieve less computation overhead. Performance analysis is carried out in Section VI. Conclusion and future work are reserved in Section VII. Table I summarizes important symbols used in this paper.

IV. STANDALONE MODE: A LIGHT-WEIGHT ONLINE UNSUPERVISED CLASSIFICATION METHOD

Accelerometer sensor retrieves three-axial acceleration data $\{x, y, z\}$ at each sampling time. Sensory readings are collected by a sliding window with a length of L and an overlap value of 50%. L is an important design merit. Shorter L may not seize the activity pattern properly, whereas the wider L would result in detection delay with extra computational workload. The overlap value is important to detect user state transitions in activity patterns.

The window at the current time is called "active frame," denoted by $f_{\{x,y,z\}}(\tau)$, where $\tau \in [t - L + 1, t]$ and t are time indexes, and L is the total number of samples for each axis window. Hence, in the case where $L/2$ number of new samples are inserted into the active frame due to the overlapping, the proposed classification method begins to operate by receiving inputs as shown in Fig. 2. The inputs are considered as two data sets: 1) the active frame; and 2) the previously active frame. The latter is denoted $f^p_{\{x,y,z\}}(\tau)$, where $\tau \in [t - 3L/2 + 1, t - L/2]$, and superscript p represents the "previous time frame."

A. Differentiation of Stationary/Nonstationary Signals

The applied method begins with normalizing each axis into unit power $\left(\text{e.g., } \hat{f}_x = f_x(t) / \left(\sqrt{\sum_{\tau=t-L+1}^{t} f_x^2(\tau)}\right)\right)$, and

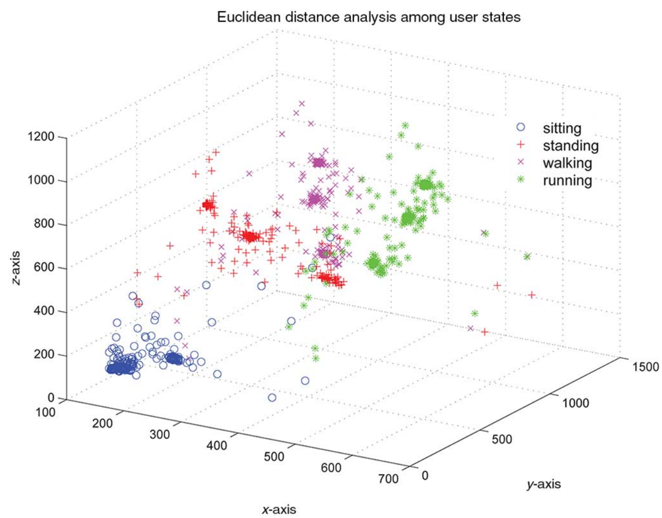

Fig. 3. Euclidean distance analysis: user state "sitting" is the reference point.

then takes cross-correlations among acceleration axis pairs in the active frame, as

$$R_{ij}(u) = \begin{cases} \sum_{\tau=0}^{L-u-1} \hat{f}_i(\tau+u)\hat{f}_j(\tau), & u \geq 0 \\ R_{ij}(-u), & u < 0 \end{cases} \tag{1}$$

where u and τ are time indexes, $i, j \in \{x, y, z\}$ and $i \neq j$.

If high correlations exist among axis pairs, i.e., $\max\{|R_{ij}|\} > \varepsilon$ where $\varepsilon \in [0.75, 1]$, user state is identified as either *sitting* or *standing*; otherwise be either *walking/running* or be in transition. Note that the applied method seeks for the highest correlation at first to specify a starting reference point, which also defines a *training data frame* for the corresponding user activity, so that the learning from future sensory samplings will be more accurate. Generally speaking, experiments show that $|R_{xy}|$ mostly satisfies the highest correlation; whereas, $|R_{xz}|$ and $|R_{yz}|$ do not. Also, the training data frame of each user activity can be updated, whenever a clear classification result is obtained from the classification of the corresponding user state.

B. Differentiation Between "Sitting" and "Standing"

Fig. 3 shows how user states occupy the Euclidean space. User state *sitting* can be differentiated from other user states easily over the Euclidean space by assigning a data set of user state *sitting* as the reference point. However, the similar conclusion cannot be made for user states *standing*, *walking*, and *running*. In other words, these three user states can be put in a same group in comparison with user state *sitting*.

The differentiation between *sitting* and *standing* relies on Euclidean distance analysis among three-axial accelerations. The reason is that the relevant data samples for these two user states are scattered distinctively over the coordinate spaces. Recall that the Euclidean distance between two random points h and q on a coordinate space is given by $|hq| = \sqrt{(h_i - q_i)^2 + (h_j - q_j)^2}$, where $i \neq j$. Hence, pairwise Euclidean distance vectors between the active frame and the training data frame for all three-axial accelerations can be calculated.

After, a radius r is defined for the training data frame that belongs to the recent recognized user state. Since the dispersion of accelerations would be distinctive to select a proper user state, the variance values of accelerations must be taken into account in order to see how far the sampling points within the training data frame are spread out from the mean. Thus, the magnitude (i.e., the norm) of a vector containing standard deviation of each axis gives out a required radius, as $r = \sqrt{\sigma_x{}^2 + \sigma_y{}^2 + \sigma_z{}^2}$. Note that the geometric mean of standard deviations could also be used to define the radius, as $r = \sqrt[3]{\sigma_x \sigma_y \sigma_z}$.

In order to identify the user state *sitting* from *standing* in the presence of Euclidean distances and defined radius, a distance-based learning function is used to cluster the user states. Range search algorithm (i.e., fixed-radius near neighbors [30]) is implemented for this purpose. It finds all points inside the pairwise Euclidean distance vectors within a radius r centered at the mean. By applying a brute-force approach, given the set H and a distance $r > 0$, all pairs of distinct points $h, q \in H$ such that $|hq| \leq r$ are found. In addition, the radius needs to be smoothed whenever a new radius value becomes available, especially at times where a perfect correlation is satisfied among axes. Hence, a better spreading circumference is maintained.

The points that stay in and out of radius r are considered to create a separation in relevant user state detections. However, to this end, the proposed method does not know yet which user state is *sitting* or *standing* due to the placement, i.e., orientation, of the smartphone. The absolute decision will be made whenever user state *walking/running* is recognized. User state *standing* lies over almost the same signal level that user state *walking/running* does.

Note that instead of range search algorithm, SVMs could be another choice to differentiate two user states from each other by using linear discrimination. SVM denotes two *user states* as binary data *classes*. The objective is to create a hyperplane that sets a rigid margin among data classes to achieve an optimal linear distance separation.

C. Detection of State Transitions

On the other hand, in case where correlations observed among axis pairs by (1) are not sufficient, i.e., $\max\{|R_{ij}|\} < \varepsilon$, where $\forall i$ and $i \neq j$, integrals of both zero-mean input frames are taken to check if both inputs are on the same signal level. Range search algorithm is applied for this purpose as well by receiving the absolute values of the both integral results, as: $|\Sigma_{\tau=t-L+1}^{t}(\tau - (t-L))(f_i(\tau) - \mu_{f_i})|$ and $|\Sigma_{\tau=t-3L/2+1}^{t-L/2}(\tau - (t-3L/2))(f_i^p(\tau) - \mu_{f_i^p})|$, where $i \in \{x, y, z\}$ and μ is the mean. The required radius is defined by the geometric mean of integral results belonging to f^p, $\sqrt[3]{\prod_{\forall i} |\Sigma_{\tau=t-3L/2+1}^{t-L/2}(\tau - (t-3L/2))(f_i^p(\tau) - \mu_{f_i^p})|}$. Finally, if the signal levels are found similar for both inputs, user state becomes either *walking* or *running*. Otherwise, we say user state is in transition, thereby the previous user state is taken as the current user state.

D. Differentiation Between "Walking" and "Running"

To differentiate *walking* and *running*, it would be reasonable to start to take the first-order differentiation of acceleration samples to exploit the contextual information better, since these predicted user states exhibit more frequently changing variations in acceleration data. Then, the first-order regression coefficient is defined by

$$\dot{f}_i(t) = \frac{(f_i(t+1) - f_i(t-1)) + 2(f_i(t+2) - f_i(t-2))}{2(1^2 + 2^2)}. \tag{2}$$

The given regression coefficient is introduced in speech signal recognition systems [31], [32] to provide temporal information about the pitches in speech signals for accuracy enhancement. The next step is to extract the mean absolute deviation (MAD) of the differentiated sensory samplings, to discern how far the points can reach out. MAD is given by

$$(\mathrm{MAD})_i = \frac{1}{L} \sum_{\tau=t-L+1}^{t} |\dot{f}_i(\tau) - \dot{\mu}_i| \tag{3}$$

where $\dot{\mu}_i$ is the mean value of the modified acceleration instance (2) for any axis. Then, range search algorithm is also applied onto points obtained from (3) to differentiate user states *walking* and *running*. When the differentiation completes, the user state that has a bigger geometric mean of all three relevant values as in (3) is marked as running; and the other one becomes *walking*.

E. Smoothness for Preventing False Truthfulness

As mentioned, user state could be in transition. The recognition process where a user state just goes into or exits from a transition would lead to having a false truthfulness, even though the cross-correlation analysis gives the opposite results by showing high correlations. In such cases, the slope of the active frame needs to be calculated as

$$\Theta_i^{\mathrm{slope}} = \arctan\left\{ \frac{\frac{4}{L}\sum_{l=3L/4+1}^{L} f_i(l) - \frac{4}{L}\sum_{l=1}^{L/4} f_i(l)}{2} \right\} \tag{4}$$

that takes the first-order difference of the mean values residing at the last and the first 25% portions of the active frame.

From (4), low angles show that the user state occurs at a same signal level, whereas high angles show that the user state transits into another one. Also, having low angles shows the evolution of the active frames still lies in the range of $|\mu_{f_i^p} \pm \sigma_{f_i^p}|$, where $\mu_{f_i^p}$ and $\sigma_{f_i^p}$ denote the mean and the standard deviation of the previously active frame. Finally, if a user transition is detected, the previous user state is taken as the current user state.

To summarize, Fig. 2 organizes the user state differentiation process from the acquired contextual data and diminishes it in the scope of DT-based classification method. This algorithm initially takes the accelerometer data stored inside the sliding window. Then, it applies the cross-correlation function through pairs of coordinate axes in the sensory data, to check for the data consistency among axes. If there is a high match observed, then the current user state becomes either *sitting* or *standing* after utilizing either range search or SVM analysis over the Euclidean space. On the other hand, user state could be *walking*, *running*, or in transition, where the inconsistency occurs after the cross-correlation checks. The differences from slope and/or integral information of two subsequent sensory data sequences expose whether user state is in transition. Low gradient in the differences reveals that the sensory data have a stable continuity on the previous signal level through axes, i.e., user state is not in transition. Finally, the differentiation between *walking* and *running* is determined by the spreading factor of the first order regression coefficients of the sensory data. Accordingly, a far distance spread within the regression coefficients indicates the user state as *running*, while the opposite indicates the user state as *walking*,

$$\{\boldsymbol{x}\} = \Big[|\mu_{\{x,y,z\}}|, \sigma_{\{x,y,z\}}, \max\{|R_{xy}|\}, \max\{|R_{xz}|\}, \max\{|R_{yz}|\}, \Theta^{\mathrm{slope}}_{\{x,y,z\}}, (\mathrm{MAD})_{\{x,y,z\}}\Big]. \tag{5}$$

Another significant merit provided by this algorithm is that a sufficient number of functions exploited from statistical/signal processing techniques are used for user state classifications. In contrast, existing classification methods do not consider this distinction either by setting simple thresholds over a specific acceleration axis while the phone placement is fixed, or by applying a large number of features to differentiate user states.

V. Assisting Mode: Enhancing Existing Classification Methods for Online Processing

The proposed method not only provides a standalone and online solution, but also it either assists pattern recognition-based classification methods (by generating training data classes), or feeds statical tool-based classification methods (by delivering input system matrices). In such cases, each user state is represented by a template data class. The properties of each class are defined by signal processing functions and primitives, which is called *feature extraction*. After, whenever a new data set comes in, diverse classification algorithms attempt to map this new data set with an existing template data class to identify the hidden user state. This process is called clustering by supervised learning, which first allows learning the mixture parameters of the stochastic distribution representing a data set, and then assumes that the new sample comes from a known distribution. In this section, we first explain the extraction of a feature vector. Then, we show how our proposed method can assist other existing classification methods for online processing.

A. Feature Extraction

Extracting features is an effective way to identify diverse characteristics from a given sensory data set. It also preserves class identity for each user state. In Section IV, features such as mean, standard deviation, cross-correlation, slope, first-order differentiation, integral, and MAD have been used. There are also many other features, e.g., angular velocity, cepstral coefficients, dc gain, energy, frequency-domain entropy, highest peaks of the power spectrum density, magnitude, peak-frequency, rms, and variance [21]–[24], [33]. However, the ultimate goal desired for online processing is to define the dimension of the feature vector $\{\boldsymbol{x}\}$ as small as possible. Therefore, we plan to construct feature vectors in this section with similar signal processing techniques. Therefore, feature vectors will be constructed as in (5) with a size of $|\mathcal{L}| = 15$. It is worth noting that the proposed method does not use every feature at a decision time. Nevertheless, existing classification methods use features at every decision time, which burdens computational complexity in online processing.

B. Pattern Recognition-Based Classification

In this section, we enhance some widely used classification methods for online processing. The introduction of these methods could be seen redundant at a first sight, whereas it is prepared concisely to have a better understanding on how their inference process of given user states takes place, and also how their system parameters are updated for online processing (see Section V-B4).

1) GMM: User state classes can be individually represented in a mixture K-Gaussian model. Any feature vector can be drawn from this model to check for which user state class (i.e., training data set) encapsulates it more. This is also called density/clustering problem, which is defined as follows. Given a set of N points in $\mathcal{D}$ dimensions, $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\} \in \mathcal{R}^{\mathcal{D}}$, find the probability density function, $f(x) \in \mathcal{F}$ on $\mathcal{R}^{\mathcal{D}}$, which is the most likely to have generated the given points.

The relevant GMM is given by

$$f(x;\theta) = \sum_{k=1}^{K} q(k,n) = \sum_{k=1}^{K} p_k \mathcal{N}(x_n; \mu_k, \sigma_k) \tag{6}$$

where $n \in N$, $k \in K$, and K-components are defined as $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\} = \{(p_1, \mu_1, \sigma_1), \ldots, (p_K, \mu_K, \sigma_K)\}$, and $\{p_k, \mu_k, \sigma_k\}$ are described as the mixing probability, the mean and the standard deviation of the model, respectively. Also,

$$\mathcal{N}(x_n; \mu_k, \sigma_k) = \frac{1}{(\sqrt{2\pi}\sigma_k)^{\mathcal{D}}} e^{-\frac{1}{2}\left(\frac{||x_n - \mu_k||}{\sigma_k}\right)^2} \tag{7}$$

is given with the properties of $\sum_{k=1}^{K} p_k = 1, p_k \geq 0$.

a) Parameter estimation: Creating training context classes: Each user state class represents a cluster that is assigned as a Gaussian model by (6) with a mean approximately in the middle of the cluster, and also with a standard deviation to show a measure of how far the cluster spreads out. Therefore, Gaussian parameters θ of each user state class needs to be estimated.

The maximum likelihood (ML) parameters $\hat{\theta}$ are estimated iteratively by using the expectation-maximization (EM) algorithm [34] in the presence of each user state class specifying feature vector, given initial estimates of $\{p_k^{(0)}, \mu_k^{(0)}, \sigma_k^{(0)}\}$ until the logarithmic likelihood function

$$\lambda(x;\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} p_k \mathcal{N}(x_n; \mu_k, \sigma_k) \tag{8}$$

converges to a local maxima, which is computed by partial differentiations of λ with respect to μ_k, σ_k, p_k, after η times iterations.

b) User state inference: Testing a given context class: For all user state classes, the corresponding joint probability of density function (pdf) of the mixture K-Gaussian model is finally given by

$$p(x_n|s) = \sum_{k=1}^{K} p_{sk} \mathcal{N}(x_n; \mu_{sk}, \Lambda_{sk}) \tag{9}$$

where $\{p_{sk}, \boldsymbol{\mu}_{sk}, \Lambda_{sk}\}$ at this time denote the weight, the mean vector, and the covariance matrix of the kth Gaussian in user state s, respectively.

For supervised learning, the number of classes and their parameters is known. Hence, with the help of the delivered training data classes by our proposed system, the GMM parameters are estimated through (8), and then (9) is derived. Whenever a new feature vector is constructed, the desired user state class s^* is identified by the cluster index satisfying ML estimation, as

$$s^* = \underset{s}{\operatorname{argmax}}\{p(x_n|s)\}. \tag{10}$$

2) k-Nearest Neighbors Search (k-NN): This algorithm assigns the nearest class set x^m, where $1 \leq m \leq k$, for a given input feature vector by defining a dissimilarity function that measures the nearness between training data set and new data points in the feature vector. The dissimilarity function is generally defined by the squared Euclidean distance as $d(x, x^m) = (x - x^m)^T(x - x^m)$. However, the Euclidean distance does not consider how the data spreads out, and may let the largest length scale among data points dominate the dissimilarity function. Therefore, the Mahalanobis distance $d(x, x^m) = (x - x^m)^T \Lambda^{-1}(x - x^m)$, where Λ is the covariance matrix, could be used to rescale all length of scales to be essentially equal. KNN classifies a feature vector x_n given the training data x^m where

$$m^* = \underset{m}{\operatorname{argmin}}\{d(x_n, x^m)\}, \quad 1 \leq m \leq k. \tag{11}$$

However, KNN becomes more computationally expensive as the dimension of the feature vector increases. Therefore, principle components analysis (PCA) is introduced [35] to reduce the dimension by replacing $\{x\}$ with a low-dimensional projection $\{y\}$. Hence, the dissimilarity function is denoted by $d(y_n, y^m)$. However, PCA may not be helpful to separate classes from each other while getting the lower dimensions. To this end, LDA is introduced [36] to reduce the dimension of feature vectors. As a result, classes are separated.

3) LDA: In supervised learning method where distinctive class information is available, the dimension of a feature vector can be reduced even when continuing to obtain a solid classification analysis. A supervised linear projection is defined as $y = W^T x$, where W is the projection matrix, $\mathcal{L}$ is the dimension of y, and $\dim\{W\} = \mathcal{D} \times \mathcal{L},\ \mathcal{L} < \mathcal{D}$. Recall that $\mathcal{D}$ denotes the dimension of feature vector x. Since each user state class s is defined with a Gaussian model $p(x_s) = \mathcal{N}(x_s; m_s, S_s)$, the Gaussian model along with linear projection turns into $p(y_s) = \mathcal{N}(y_s; \mu_s, \sigma_s^2)$, where $\mu_s = W^T m_s$ and $\sigma_s^2 = W^T S_s W$.

The most efficient projection matrix creates a minimal overlap among the projected distributions, which can be achieved when the projected means are maximally separated, and the projected variance is not large enough. Therefore, the objective function is known via maximizing the Fisher criterion [37], as

$$\frac{(\mu_s - \mu_{\acute{s}})^2}{\pi_s\sigma_s^2 + \pi_{\acute{s}}\sigma_{\acute{s}}^2} \rightarrow \mathcal{F}(W) = \frac{W^T(m_s - m_{\acute{s}})(m_s - m_{\acute{s}})^T W}{W^T(\pi_s\sigma_s^2 + \pi_{\acute{s}}\sigma_{\acute{s}}^2)^T W} = \frac{W^T A W}{W^T B W} \tag{12}$$

where π_s is the fraction of data set in class s, $s \neq \acute{s}$, $A = (m_s - m_{\acute{s}})(m_s - m_{\acute{s}})^T$, and $B = \pi_s\sigma_s^2 + \pi_{\acute{s}}\sigma_{\acute{s}}^2$.

In case of more than one dimension and/or more than two classes exist, Fisher's projection method in (12) is generalized by Canonical Variates method that takes $p(x) = \mathcal{N}(x; \mu_s, \Lambda_s)$ and replaces it with $p(y) = \mathcal{N}(W^T x; W^T \mu_s, W^T \Lambda_s W)$. Then, the following matrices are defined.

- Between class scatter: $A = \sum_{\forall s} n_s(\mu_s - \mu)(\mu_s - \mu)^T$ where μ is the mean of all data set, and n_s is the number of data points in class s.
- Within class scatter: $B = \sum_{\forall s} n_s B_s$, where $B_s = \frac{1}{n_s}\sum_{u=1}^{n_s}(x_u^s - \mu_s)(x_u^s - \mu_s)^T = \Lambda_s$.

By assuming that B is invertible, the Cholesky factor $\tilde{B}$ is defined by $\tilde{B}^T\tilde{B} = B$. In addition, defining $\tilde{W} = \tilde{B}W$ yields to have $W = \tilde{B}^{-1}\tilde{W}$. Then, the latest Fisher criterion becomes

$$\mathcal{F}(\tilde{W}) = \frac{(\tilde{B}^{-1}\tilde{W})^T A \tilde{B}^{-1}\tilde{W}}{(\tilde{B}^{-1}\tilde{W})^T \tilde{B}^T\tilde{B}\tilde{B}^{-1}\tilde{W}} = \frac{\tilde{W}^T\tilde{B}^{-T}A\tilde{B}^{-1}\tilde{W}}{\tilde{W}^T\tilde{W}} \tag{13}$$

which subjects to $\tilde{W}^T\tilde{W} = I$.

Since it is symmetric, the term $\tilde{B}^{-T}A\tilde{B}^{-1}$ in (13) turns into a special case of the real-valued Eigen-decomposition form, as: QEQ^T, called "Schur decomposition," where E is a diagonal matrix that contains the eigenvalues. Hence, $\tilde{W}$ is set to hold relevant eigenvectors. Finally, the projection matrix becomes

$$W = \tilde{B}^{-1}\tilde{W}. \tag{14}$$

To this end, in order to differentiate user states, the newest (i.e., lower dimensional) form of data classes and feature vectors are modeled by (9) and (10).

4) Online Processing: Dynamic Training/Supervisory Class Update: For an online classification algorithm, it is critical to reduce the computational burden and stay away from a large amount of data manipulations. The computational complexity requires $\mathcal{O}(\eta\mathcal{L}\mathcal{D}^2)$, $\mathcal{O}(\mathcal{L}^2\mathcal{D})$, and $\mathcal{O}((\mathcal{L}+c)^2\mathcal{D})$ times for GMM, k-NN, and LDA with Schur decomposition algorithms, respectively. Here, η is the number of iterations during EM algorithm in GMM, and c is the total number of user state classes. In this sense, the parameters related to GMM, k-NN, and exceptionally LDA algorithms need to be dynamically updated in an efficient way, rather than just computing relevant values all over again when either new training data samples are inserted into an existing class, or a new data class is added/deleted. Especially, matrix multiplications, which take $\mathcal{O}(c\mathcal{L}\mathcal{D})$ time, need to be optimized during the update process.

Here is the suggested update for supervised classification algorithms.

- Adding new data to an existing class i: ($\hat{n}_s$: the number of added training samples)
  - Common properties for all classification methods:
    - $\hat{\mu}_s = \mu_s + \Delta_{\mu_s}$.
    - $\Delta_{\mu_s} = ((\sum_{u=n_s+1}^{n_s+\hat{n}_s} x_u^s) - \hat{n}_s\mu_s)/(n_s + \hat{n}_s)$.
  - Only additional for LDA:
    - $\hat{\mu} = \frac{\left((n_s+\hat{n}_s)\mu + \sum_{\forall s}\sum_{u=n_s+1}^{n_s+\hat{n}_s} x_u^s\right)}{(\sum_{\forall s} n_s+\hat{n}_s)}$.
    - $\hat{A} = \sum_{\forall s}(n_s + \hat{n}_s)(\hat{\mu}_s - \hat{\mu})(\hat{\mu}_s - \hat{\mu})^T$.
    - $\hat{B}_s = \frac{1}{n_s+\hat{n}_s}\sum_{u=1}^{n_s+\hat{n}_s}(x_u^s - \hat{\mu}_s)(x_u^s - \hat{\mu}_s)^T$.
    - $\hat{B} = \sum_{\forall s}\left(n_s B_s + n_s\Delta_{\mu_s}\Delta_{\mu_s}^T + \cdots + \sum_{u=n_s+1}^{n_s+\hat{n}_s}(x_u^s - \hat{\mu}_s)(x_u^s - \hat{\mu}_s)^T\right)$.
- Adding/deleting a class $\{s^*\}$:
  - Common properties for all classification methods:
    - create/delete $\mu_{\{s^*\}}$ and $\Lambda_{\{s^*\}}$.
  - Only additional for LDA:
    - $\hat{B} = B \pm n_{\{s^*\}}B_{\{s^*\}}$.
    - $\hat{A} = A + n\Delta_\mu\Delta_\mu^T \pm n_{\{s^*\}}(\mu_{\{s^*\}} - \hat{\mu})(\mu_{\{s^*\}} - \hat{\mu})^T$.
    - $\Delta_\mu = \hat{\mu} - \mu = \pm n_{\{s^*\}}(\mu_{\{s^*\}} - \mu)/(n \pm n_{\{s^*\}})$.

Note that after the update process completes, only for LDA algorithm, the linear projection matrix needs to be recomputed.

C. Statistical Tool-Based Classification: HMMs

HMMs are only introduced as a statistical tool-based classification in this section. HMM-based human postural behavior and activity detections are mostly used statistical tools in HAR objected applications. In HMMs, a system parameter called "observation emission matrix" is represented with the help of (9) by

$$b_{s,O_t} = p(O_t/q_t = s) = \sum_{k=1}^{K} p_{sk}\mathcal{N}(O_t; \mu_{sk}, \Lambda_{sk}) \tag{15}$$

where s, O_t, and q_t are a user state (i.e., a data class), an instant observation (i.e., a feature vector), and a user state instance in a sequence, respectively. The traditional model only considers that observations (i.e., input feature vector) are Gaussian, and it draws an instant feature vector from cross-relationships between observations and user states to decide the most likely inference of an instant user state. The user state inference, according to (10), is done by checking for Gaussian membership of observations, and it is then selected as the suitable user state outcome regulated by majority voting in the classification.

On the other hand, HMMs like Baum–Welch method (forward-backward algorithms) [38] utilize heavy computations, while evolving its chain that produces user state inferences in decision epochs. Therefore, to improve HMMs for online processing, a similar system matrix like the one in (15) can be constructed. This matrix is given by

$$b_{s,O_t} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{\text{in}}^{\text{Euc.}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + R_{\text{out}}^{\text{Euc.}} \right) + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{\text{in}}^{\text{MAD}} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} R_{\text{out}}^{\text{MAD}} \right) \quad (16)$$

where $\otimes$ is the Kronecker product, $R_{\text{in/out}}^{\text{Euc.}}$ is the percentage of points that stays in and out of radius within the fixed search algorithm (for Euclidean distance analysis to differentiate user states *sitting*, $\{\text{in}\}$), and *standing*, $\{\text{out}\}$. $R_{\text{in/out}}^{\text{MAD}}$ is the same approach, but now it is based on MAD analysis to differentiate user states *walking*, $\{\text{in}\}$, and *running*, $\{\text{out}\}$.

In addition, another important system parameter used in HMMs, i.e., the user state transition matrix, is also defined by using our proposed method as

$$a_{s,\acute{s}} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix} \quad (17)$$

where s and $\acute{s}$ denote user states, ordered as *sitting, standing, walking*, and *running*. In most studies, this matrix is formed by user interventions after examining a full data record of their activities at offline, and specifying a general trend of all users with respect to specific user states. However, in our proposed method, this matrix is held *fixed with equal probabilities* as default. It is so because that it is not possible to give a specific formation to present user state transitions. Therefore, we focus on how to create the observation emission matrix. According to the defined user state transition matrix, it is not possible to transit from *sitting* to *running*, from *standing* to *running*, or vice versa. The rest of the elements in the matrix has equal probability with respect to other state transitions.

VI. Performance Evaluation

To demonstrate the effectiveness of the proposed classification method, experiments are carried out by using Blackberry RIM Storm II 9550 smartphone as target device. Blackberry Java 7.1 SDK is used for programming purpose of implementations, and Eclipse is used as software development tool. Storm II consists of a three-axial accelerometer sensor, named ADXL346 from analog devices. The experiments are performed by eight different individuals with around 16 h accelerometer data recording analysis in total *per each method* (see Table II). For instance, Fig. 4 shows a 10-min recording of the collected sensory readings from the sensor, and it draws the track of user state recognitions by our proposed method, with respect to changing contexts in the readings.

The target device is considered to be put in trousers pocket. A change in orientation of the device, such as rotation, is an important design drawback for most of classification algorithms, especially for those which solely rely on exploiting a specific axis information. Since the sensor is not placed fixed, it would produce some distortion over acceleration axes. Upward or downward position of the device causes x-axis flipped to y-axis, or vice versa. Therefore, the device is placed fixed in many studies. Any change in orientation could cause false truthfulness to happen in activity recognitions. The proposed classification method in this paper is not affected much by device rotations, since it does not depend on fixed threshold values in a context recognition process. As the worst-case scenario, whenever the device changes its rotation, the update process will handle the adaptation problem as long as the rotation does not occur all the time. For example, if a rotation occurs, user state *sitting* might start being selected as *standing*, and this would give a false truthfulness to differentiate user states, as long as user state *walking/standing* is not perceived. It is worth noting that in our proposed algorithm, user states *walking/standing* give out user state *standing* since their data signal levels flow similarly.

Accelerometer sensor is sampled at $f_s = \{100, 50, 25\}$ Hz, and samplings are windowed with $L = 3f_s$ and 50% overlap value. Exceptionally, when $f_s = 100$ Hz, L could be $2f_s$ since lower sampling frequencies cannot resolve inference problem when window size is not adequate. The proposed method infers user states from the obtained contextual information through the accelerometer sensor with a perfect accuracy, as shown in

TABLE II
Diversity in Participants Who Involve Experiments

Age	Gender	Number per gender	Data length per subject (h)
18–30	Male	2	2
	Female		
30–40	Male	1	2
	Female		
>40	Male	1	2
	Female		

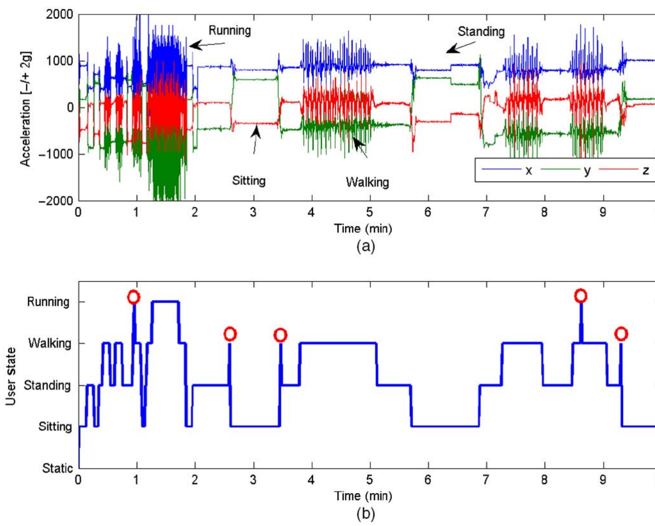

Fig. 4. Context inference from the accelerometer sensor: (a) a 10-min recording of three-axial acceleration signals while user posture changes; (b) the corresponding user state representations before smoothing is applied.

TABLE III
CONFUSION MATRIX 1: USER STATE RECOGNITION UNDER DIFFERENT CLASSIFICATION METHODS AT 100 HZ SAMPLING

Method at 100 Hz		a	b	c	d
DT	a	98.2	1.8	0	0
	b	1.4	94.5	4.1	0
	c	0	2.7	93.8	3.5
	d	0	0	8.3	91.7
GMM	a	93.8	4.4	0.6	0
	b	9.6	86.2	4.2	0
	c	0.6	3.7	89.5	6.1
	d	0	0	10.1	89.9
k-NN	a	94.5	4.4	1.1	0
	b	2.1	89.7	8.2	0
	c	0	6.6	82.1	11.3
	d	0	0	9.8	90.2
LDA	a	93.2	6.2	0.6	0
	b	8.5	83.5	8	0
	c	0.5	6.2	81.7	11.6
	d	0	3.3	12.4	84.3
HMM	a	95.3	4.7	0	0
	b	4.1	92.4	3.5	0
	c	0	1.8	91.5	6.7
	d	0	0	9.7	90.3

a, sitting; b, standing; c, walking; d, running.

TABLE IV
CONFUSION MATRIX 2: DT UNDER DIFFERENT SAMPLING FREQUENCIES

DT		a	b	c	d
@50 Hz	a	95.2	4.8	0	0
	b	3.2	93.1	3.7	0
	c	0	3.8	86.5	9.7
	d	0	0	15.9	84.1
@25 Hz	a	93.1	6.9	0	0
	b	5.4	88.3	6.3	0
	c	0	9.4	78.3	12.3
	d	0	0	20.7	79.3

a, sitting; b, standing; c, walking; d, running.

Fig. 4. In addition, a user's quick movements during state transitions may lead to false statements in the recognition process, especially any user transition between sitting/standing/walking might be detected as walking, or any user transition from/to walking might be detected as running. In such cases, a basic *smoothing* technique is applied, by taking a majority voting scheme, where a sliding window with a specific length of user state history is used to prevent from having false truthfulness. Hence, the red circle in Fig. 4(b) is corrected, since the preceding and proceeding user states are different from what is perceived.

As shown in Fig. 2, user state recognition analysis is examined along with (10), (11), (14), and (16) for each classification method. For each method, results with different sampling frequencies are evaluated in a form of confusion matrices (see Tables III and IV), whose rows and columns specify the desired and detected user states, respectively, and whose values show the match ratios in a user state recognition process. Table III shows the confusion matrix for user state recognitions under different classification methods at 100 Hz from accelerometer sensor samplings. The proposed classification method, referred as "DT," achieves a great differentiation in user states

TABLE V
BATTERY LIFETIME WITH RESPECT TO CONSTANT APPLICATION USE UNDER DIFFERENT CLASSIFICATION ALGORITHMS WITHOUT UPDATE

Method	Battery lifetime (h)
DT	8.5
GMM	3
k-NN	5.5
LDA	4
HMM	6

recognitions, since it well analyzes the acceleration signals and fully exploits sufficient features to make such differentiations in user states. Other methods succeed with very reasonable truthfulness. This is because that the online processing allows them to update training classes of the clustering problem, instead of having fixed data references all the time. Furthermore, our proposal also achieves certain level of adaptability toward some heterogeneities observations over acceleration signals, such as distortion, due to the rapid movement or change in orientation. Only LDA algorithm seems to have worse outcomes, which might result from the size of used feature vector. That is, since the feature vector in LDA is set to a length of 15, it may not resolve the clustering problem after size reduction due to projection. It is worth noting that most approaches mentioned in Section II have a very large size of feature vectors, e.g., size of over 40 features. In a summary, existing classification methods fall behind with the proposed one, even if they benefit from all signal processing elements utilized in a feature vector while differentiating user state classes.

Furthermore, Table IV shows how the proposed classification method responds under slower sampling frequencies. Our proposal still achieves a solid differentiation between *sitting* and *standing*, whereas there exist some confusions between *walking* and *running*. Having lower number of samplings and variant signal characteristics of actions such as *walking/running*, it would be difficult to have a pure data frame for each user state representation. Hence, clustering problem becomes more challenging to be resolved.

On the other hand, in order to explore the impact of each classification algorithm on the battery depletion, an application is implemented on the target device. It runs from the point where 1400 mAh smartphone battery is fully loaded, until it totally depletes. Only one classification algorithm without update process is employed at each experiment. Experiments with update process are not considered in battery depletion analysis, since different users may exhibit various activity patterns, and this makes update process take place in different times with many repetitions in the presence of high-accurate detections. While applications are running, the device is left in the standby mode, and only connected to a 3G network. Also, the accelerometer sensor is set to the aggressive sampling mode, where $f_s = 100$ Hz and $\mathrm{dc} = 100\%$. Results are given in Table V. We observe that the proposed method achieves the longest battery lifetime. It is because that the proposed contextual inference model fully exploits characteristics of accelerometer signals and applies sufficient number of signal processing features as needed. It also does not apply all features at the same time like

TABLE VI
COMPARISON OF CLASSIFICATION METHODS WITH RESPECT TO AVERAGE PROCESS TIME

Method	Process time (ms)		Update included
	Mean	Std.	
DT	5	0.9	-
	6	1.0	✓†
GMM	12	0.6	-
	15	2.1	✓†
	22	3.0	✓‡
k-NN	7	0.7	-
	8.5	1.3	✓†
	12	1.5	✓‡
LDA	8.5	0.5	-
	20	2.7	✓†
	27	3.2	✓‡
HMM	6.4	0.6	-
	8.3	1.2	✓†

†, proposed update; ‡, regular update.

the other classification methods. Therefore, LDA and GMM together have shorter battery lifetime, since they both first create feature vectors after sensory data acquisition, and test these vectors with high-dimensional user state class matrices to check the membership of vectors in correspondence with any user state classes.

In addition to battery lifetime comparison, Table VI is provided to well examine and compare classification algorithms in terms of computational complexity during context inference with/without update process taken place on their system parameters. As can be seen, our proposed method proves its lightweightiness property by having a lower time span to process sensory data until inferring desired context as outcome, and it also outperforms other methods even if update is applied. To mention the other classification methods, LDA takes less process time than GMM while inferring context thanks to its complexity reducing method. However, it uses GMM-based training data sets and applies projection on them to reduce complexity in computations. Therefore, update time of LDA's system parameters is longer than GMM's. k-NN and HMM together show very close results in computation time. However, it is normally expected from HMM to have a longer process time span since training of system parameters are needed in update process. Recall that our proposed system creates the observation emission matrix in HMM, and that pulls process time down.

Therefore, the proposed method achieves solid user state differentiations under variant sampling frequencies by showing a more than 90% overall accuracy. Also, it shows a 10% overall enhancement for each user state at 100 Hz, when comparing with the widely used GMM, k-NN, LDA, and HMM-based classification methods. In addition, the proposed method infers user states from the obtained context through the accelerometer sensor with almost a perfect accuracy at 100 Hz, as an example shown in Fig. 4(b). For slower sampling frequencies, the method still makes high accurate differentiations in user states. On the other hand, it outperforms other classification methods by achieving less computational complexity and more battery lifetime.

VII. CONCLUSION

This paper has proposed an online classification method to detect users' postural actions (e.g., *user states, as: sitting, standing, walking, and running*) by using the smartphone accelerometer sensor. Specifically, a novel DT-based unsupervised learning method is developed. It effectively exploits the context provided by acceleration signals with an adequate number of signal processing techniques, and the fixed-radius neighbors search algorithm. Therefore, there is no need to create high-dimensional feature vectors to feed into the pattern recognition algorithms to cluster the desired user state classes. This allows our method to be lightweight. Furthermore, the proposed method does not need any *priori* information about user state classes. It also provides training data classes to assist other widely used existing classification methods (e.g., GMM, k-NN, LDA, and HMM). Finally, we improve these computationally heavyweight methods for online processing, by providing a flexible update method on their system parameters. Extensive experiments are performed successfully through a smartphone accelerometer sensor. Results show that the proposed method achieves solid user state differentiations under variant sampling frequencies. That is, it achieves an above 90% overall accuracy at each sampling frequency, and also outperforms GMM, k-NN, LDA, and HMM-based classifications with both an 10% overall enhancement for each user state at 100 Hz, and with an extended battery lifetime.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] Z. Sheng *et al.*, "A survey on the ietf protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[3] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, "Body area networks: A survey," *Mobile Netw. Appl.*, vol. 16, no. 2, pp. 171–193, Apr. 2011.

[4] A. V. Vasilakos, "Special issue: Ambient intelligence," *Inf. Sci.*, vol. 178, no. 3, pp. 585–587, 2008.

[5] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A survey on ambient intelligence in healthcare," *Proc. IEEE*, vol. 101, no. 12, pp. 2470–2494, Dec. 2013.

[6] E. Jovanov and A. Milenkovic, "Body area networks for ubiquitous healthcare applications: Opportunities and challenges," *J. Med. Syst.*, vol. 35, no. 5, pp. 1245–1254, Oct. 2011.

[7] E. M. Berke, T. Choudhury, S. Ali, and M. Rabbi, "Objective measurement of sociability and activity: Mobile sensing in the community," *Ann. Family Med.*, vol. 9, no. 4, pp. 344–350, 2011.

[8] K. G. Stanley and N. D. Osgood, "The potential of sensor-based monitoring as a tool for health care, health promotion, and research," *Ann. Family Med.*, vol. 9, no. 4, pp. 296–298, 2011.

[9] I. Sabek, M. Youssef, and A. Vasilakos, "Ace: An accurate and efficient multi-entity device-free wlan localization system," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, pp. 261–273, Feb. 2015.

[10] H. Bruce *et al.*, "An extensible sensor based inferencing framework for context aware applications," in *Proc. IEEE 10th Int. Conf. Comput. Inf. Technol.*, 2010, pp. 2878–2883.

[11] B. S. Jensen, J. E. Larsen, K. Jensen, J. Larsen, and L. K. Hansen, "Estimating human predictability from mobile sensor data," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Aug. 2010, pp. 196–201.

[12] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proc. Conf. Innov. Appl. Artif. Intell.*, 2005, pp. 1541–1546.

[13] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Pers. Ubiq. Comput.*, vol. 14, no. 7, pp. 645–662, Oct. 2010.

[14] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 849–860, 2008.

[15] S. Biswas and M. Quwaider, "Body posture identification using hidden Markov model with wearable sensor networks," in *Proc. ICST 3rd Int. Conf. Body Area Network (BodyNets'08)*, 2008, Art. ID 19.

[16] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *Proc. IEEE 30th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBS)*, Aug. 2008, pp. 5250–5253.

[17] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 48–53, Jan./Mar. 2010.

[18] Q. Shi, L. Wang, L. Cheng, and A. Smola, "Discriminative human action segmentation and recognition using semi-markov model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2008, pp. 1–8.

[19] S. Zhong and J. Ghosh, "Hmms and coupled hmms for multi-channel eeg classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2002, vol. 2, pp. 1154–1159.

[20] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using ar model representation and svm," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2008, vol. 4, pp. 2245–2250.

[21] E. M. Tapia *et al.*, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *Proc. IEEE Int. Symp. Wearable Comput.*, 2007, pp. 1–4.

[22] J. Parkka *et al.*, "Activity classification using realistic data from wearable sensors," *IEEE Trans. Inf. Tech. Biomed.*, vol. 10, no. 1, pp. 119–128, Jan. 2006.

[23] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[24] M. Ermes, J. Parkka, and L. Cluitmans, "Advancing from offline to online activity recognition with wearable sensors," in *Proc. IEEE 30th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBS)*, Aug. 2008, pp. 4451–4454.

[25] D. Peebles, H. Lu, N. D. Lane, T. Choudhury, and A. T. Campbell, "Community-guided learning: Exploiting mobile sensor users to model human behavior," in *Proc. AAAI*, 2010.

[26] P. Siirtola and J. Rning, "Recognizing human activities user-independently on smartphones based on accelerometer data," *Int. J. Interact. Multimedia Artif. Intell. (IJIMAI)*, vol. 1, no. 5, pp. 38–45, 2012.

[27] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Proc. Int. Conf. Ubiq. Intell. Comput.*, 2010, pp. 548–562.

[28] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *Proc. 16th Int. Symp. Wearable Comput.*, 2012, pp. 17–24.

[29] J. Yang, H. Lu, Z. Liu, and P. Boda, "Physical activity recognition with mobile phones: Challenges, methods, and applications," in *Multimedia Interaction and Intelligent User Interfaces*. New York, NY, USA: Springer, 2010, pp. 185–213.

[30] J. L. Bentley, D. F. Stanat, and E. H. Williams, Jr., "The complexity of finding fixed-radius near neighbors," *Inf. Process. Lett.*, vol. 6, no. 6, pp. 209–212, 1977.

[31] F. Lefvre, C. Montaci, and M.-J. Caraty, "On the influence of the delta coefficients in a HMM-based speech recognition system," in *Proc. Int. Speech Sci. Tech. Conf.*, Dec. 1998.

[32] O. Thomas *et al.*, "Wearable sensor activity analysis using semi-Markov models with a grammar," *Pervasive Mobile Comput.*, vol. 6, no. 3, pp. 342–350, Jun. 2010.

[33] Y. Hanai, J. Nishimura, and T. Kuroda, "Haar-like filtering for human activity recognition using 3d accelerometer," in *Proc. IEEE 13th Digit. Signal Process. Workshop, 5th IEEE Signal Process. Educ. Workshop*, Jan. 2009, pp. 675–678.

[34] J. A. Bilmes *et al.*, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models," *Int. Comput. Sci. Inst.*, vol. 4, no. 510, p. 126, 1998.

[35] M. Li *et al.*, "Multimodal physical activity recognition by fusing temporal and cepstral information," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 4, pp. 369–380, Aug. 2010.

[36] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. Int. Workshop Wearable Implantable Body Sensor Netw.*, Apr. 2006, pp. 4–116.

[37] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2011.

[38] L. R. Welch, "Hidden Markov models and the Baum-Welch algorithm," *IEEE Inf. Theory Soc. Newslett.*, vol. 53, no. 4, pp. 10–13, 2003.

Özgür Yürür (M'13) received the Ph.D. degree in electronics engineering and computer engineering (double major) from the Gebze Institute of Technology, Kocaeli, Turkey, in 2008, the M.S.E.E. degree in electrical engineering from the University of South Florida (USF), Tampa, FL, USA, in 2010, and is currently working toward the Ph.D. degree in electrical engineering at USF.

His research interests include mobile sensing, ubiquitous sensing, mobile computing, machine learning, and energy efficient optimal sensing policies in wireless networks.

Chi Harold Liu (M'10) received the Ph.D. degree from Imperial College, London, U.K., and the B.Eng. degree from Tsinghua University, Beijing, China.

He is currently a Full Professor with the School of Software, Beijing Institute of Technology, Beijing, China. He is also the Director of the IBM Mainframe Excellence Center, Beijing, China, the Director of the IBM Big Data Technology Center, Beijing, China, and the Director of the National Laboratory of Data Intelligence for China Light Industry, Beijing, China. Before moving to academia, he joined IBM Research, Beijing, China, as a Staff Researcher and Project Manager, after working as a Postdoctoral Researcher with Deutsche Telekom Laboratories, Berlin, Germany, and a Visiting Scholar with the IBM T. J. Watson Research Center, New York, NY, USA. He has authored more than 60 prestigious conference and journal papers and owned more than 10 EU/U.S./China patents. His research interests include the Internet of Things (IoT), big data analytics, mobile computing, and wireless *ad hoc*, sensor, and mesh networks.

Dr. Liu is a Member of the ACM. He serves as an Editor for the *KSII Transactions on Internet and Information Systems* and is a Book Editor for four books published by the Taylor & Francis Group, USA. He served as a Consultant with Asian Development Bank, Manila, Philippines, Bain & Company, Boston, MA, USA, and KPMG, Beijing, China, and a Peer Reviewer with the Qatar National Research Foundation, Doha, Qatar, and National Science Foundation, Beijing, China. He also has served as the General Chair of the IEEE SECON'13 workshop on IoT Networking and Control, the IEEE WCNC'12 workshop on IoT Enabling Technologies, and the ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He was the recipient of the Distinguished Young Scholar Award in 2013, the IBM First Plateau Invention Achievement Award in 2012, and the IBM First Patent Application Award in 2011. He was interviewed by EEWeb.com as the Featured Engineer in 2011.

Wilfrido Moreno received the B.S.E.E., M.S.E.E., and Ph.D. degrees from the University of South Florida (USF), Tampa, FL, USA.

He is currently a Professor with the Department of Electrical Engineering, USF, and the Director of the Ibero-American Science and Technology Educational Consortium, USF (ISTEC-USF). In addition, he has served as a Director of the R&D Initiative for ISTEC Inc. since the fall of 2003. He has supervised industrial projects applying system of systems engineering methodologies. He has authored over 100 technical papers, has participated in over 150 International conferences and seminars around the World, and has organized over 50 hands-on workshops in the area of electronic design automation, systems engineering and controls, nano-electronics, and digital signal processing. His research interests include system integration by providing "off-the-shelf" hardware/software solutions to industrial applications, digital signal processing, communications, industrial power and controls, nano/micro-electronics, biomedical engineering and multimedia solutions, and optimal sensing.

Dr. Moreno is a Founding Member of the former Center for Microelectronics Research (CMR), Tampa, FL, USA, which is currently the Nanotechnology Research & Education Center (NREC), Tampa, FL, USA.