Sensors **2014**, *14*, 22500-22524; doi:10.3390/s141222500

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Long-Term Activity Recognition from Wristwatch Accelerometer Data [†]

Enrique Garcia-Ceja *, Ramon F. Brena, Jose C. Carrasco-Jimenez and Leonardo Garrido

Tecnológico de Monterrey, Campus Monterrey, Av. Eugenio Garza Sada 2501 Sur,
Monterrey 64849, Mexico; E-Mails: ramon.brena@itesm.mx (R.F.B.);
jccarrasco05@gmail.com (J.C.C.-J.); leonardo.garrido@itesm.mx (L.G.)

[†] Expanded conference paper based on "Long-Term Activities Segmentation Using Viterbi Algorithm with a k-Minimum-Consecutive-States Constraint", the 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014).

* Author to whom correspondence should be addressed; E-Mail: A00927248@itesm.mx;
Tel.: +52-81-8358-2000 (ext. 5246).

External Editor: Panicos Kyriacou

Received: 30 July 2014; in revised form: 18 October 2014 / Accepted: 14 November 2014 / Published: 27 November 2014

Abstract: With the development of wearable devices that have several embedded sensors, it is possible to collect data that can be analyzed in order to understand the user's needs and provide personalized services. Examples of these types of devices are smartphones, fitness-bracelets, smartwatches, just to mention a few. In the last years, several works have used these devices to recognize simple activities like running, walking, sleeping, and other physical activities. There has also been research on recognizing complex activities like cooking, sporting, and taking medication, but these generally require the installation of external sensors that may become obtrusive to the user. In this work we used acceleration data from a wristwatch in order to identify long-term activities. We compare the use of Hidden Markov Models and Conditional Random Fields for the segmentation task. We also added prior knowledge into the models regarding the duration of the activities by coding them as constraints and sequence patterns were added in the form of feature functions. We also performed subclassing in order to deal with the problem of intra-class fragmentation, which arises when the same label is applied to activities that are conceptually the same but very different from the acceleration point of view.

Keywords: activity recognition; long-term activities; accelerometer sensor; CRF; HMM; Viterbi; clustering; subclassing; watch; context-aware

1. Introduction

Human activity recognition is an important task for ambient intelligent systems. Being able to recognize the state of a person can provide us with valuable information that can be used as input for other systems. For example, in healthcare, fall detection can be used to alert the medical staff in case of an accident; in personal assistant applications, the current activity could be used to improve the recommendations and reminders. For example, in [1] a framework for activity inference is proposed, which is based on the idea that it is possible to classify activities from the handled artifacts used to perform each activity. They presented a practical case in a nursing home to characterize the activities that caregivers perform in providing healthcare of elders with restricted mobility. Han *et al.* [2] proposed a healthcare framework to manage lifestyle diseases by monitoring long-term activities and reporting irregular and unhealthy patterns to a doctor and a caregiver.

With the development of wearable devices that have several embedded sensors, it has become possible to collect different types of data that can be analyzed in order to understand the user's needs and provide personalized services. Examples of these types of devices are smartphones, fitness bracelets [3], smartwatches [4], *etc.* With the miniaturization of sensors, it is now possible to collect data about acceleration, humidity, rotation, position, magnetic field, and light intensity with small wearable devices. In recent years, simple human activity recognition has been achieved successfully; however, complex activity recognition is still challenging and remains an active area of research. Generally, simple activities do not depend on the context, *i.e.*, they can exist by themselves and they last for only a few seconds. Examples of this type of activities are: running, walking, resting, sitting, *etc*. More complex and long-term activities are composed of a collection of simple activities and may include additional contextual information like time of the day, spatial location, interactions with other people and objects. Examples of this type of activities include: cooking, sporting, commuting, taking medication, among other types of activities. The recognition of these activities generally requires more sensors and a fixed infrastructure (e.g., video cameras, RFID tags, several accelerometers, magnetic sensors). In this work we focus on the problem of recognizing sequential long-term activities from wristwatch accelerometer data and the problem can be stated as follows: Given a sequence of accelerometer data recorded from a wristwatch, the task is to recognize what long-term activities were performed by the user and the order in which they occurred, *i.e.*, their segmentation. In this work, we focused on activities of daily living, such as shopping, exercising, working, taking lunch, *etc.*, since they are useful to living independently [5]. This information could also be used by health and wellbeing applications such as Bewell [6] in order to assist individuals in maintaining a healthy lifestyle. Another potential application is that the activities could be further characterized in order to understand their quality. For example, when doing exercise, we may want to know if the user is guarding against injuries or performing the activity with confidence. Aung *et al.* [7] and Singh *et al.* [8] have discussed the importance of detecting the quality of activities to provide a more tailored support or feedback in mental and physical rehabilitation.

To be able to recognize the long-term activities, first we decomposed the accelerometer data into a sequence of *primitives* [9]; thus, each long-term activity is represented as a string where each symbol is a *primitive*. This process is called *vector quantization* ([10], pp. 122–131) and will be described in Section 5. A *primitive*, conceptually represents a simple activity, e.g., running, walking, *etc.*, but there is not necessarily a one-to-one mapping between the primitives and activities that exist in our "human vocabulary". These *primitives* are automatically discovered from the data [11]. The rationale for extracting these *primitives* is that the data can be represented as a sequence of characters, and thus, sequence patterns specific to each of the activities can be looked for in order to increase the discrimination power between different activities. This type of representation is also suitable to perform activity prediction [12] (which will be left as future work). We used the Viterbi algorithm [13,14] on a Hidden Markov Model (HMM) ([10], Chapter 6) and on a Conditional Random Field (CRF) [15] to perform the segmentation and compared the results.

Within the HMMs and CRFs context, the sequence of *primitives* can be seen as the sequence of observations. For each observation we want to know the class (the activity) that generated it. The Viterbi algorithm is a dynamic programming method to find the best state sequence (in this case, the sequence of activities) that generated the observations (*primitives*) according to some optimality criterion. We used a modified version of this algorithm to add a constraint that takes into account the activities minimum lifespan. For example, if we know that the *shopping* activity takes at least 10 min, we can rule out state sequences that are shorter than 10 min. In our previous work of activity segmentation with an HMM, we called this a k-minimum consecutive states constraint [16]. In this work, we will also add information about sequence patterns found in the data by using a CRF.

When modeling the activities with an HMM, it is difficult to incorporate prior knowledge in the form of sequence patterns because HMMs have difficulty in modeling overlapping, non-independent features [17]. A Conditional Random Field can be thought of as a more general Hidden Markov Model. With CRFs it is possible to add arbitrary features even if they overlap. We identified recurrent patterns that occur within each activity and coded them as features in CRFs.

In this paper we present an extension of our previous work about long-term activities segmentation [16] with the following additions: (1) the use of real data collected with a wristwatch from two subjects (21 days of data) instead of using simulated data; (2) the use of Conditional Random Fields in order to include more information into the models by finding sequence patterns; (3) the data and code were made publicly available to facilitate the reproduction of the results; (4) the idea of subclassing was introduced in order to deal with the problem of intra-class fragmentation. We also make use of clustering quality measures [18] in order to automatically find good subclassing groups. This problem arises during the activity labeling process. For example, a user could label an activity as *having dinner* but sometimes he may use cutlery (for meat, soups, salads, *etc.*) and sometimes he may use his hands (for hamburgers, pizza, *etc.*). In this case, the activity label is the same for both scenarios even though both may vary widely from the acceleration point of view, making the activity *having dinner* implicitly fragmented into two classes. This phenomenon can introduce noise to the model and decrease the recognition accuracy. In Section 7 we will describe the process to deal with the intra-class fragmentation problem by automatically finding possibly fragmented classes and *subclassing* them.

This paper is organized as follows. Section 2 presents related works on activity recognition. Section 3 presents the background theory of Hidden Markov Models and Conditional Random Fields. In Section 4 we present the data collection process and the preprocessing steps. Section 5 details how the sequences of primitives are generated from the raw accelerometer data. Section 6 explains the features used for the CRF, which includes finding the sequence patterns. Section 7 describes the intra-class fragmentation problem and the proposed method to reduce its effects. In Section 8 we describe the experiments and present the results. Section 9 directs to the sources for downloading the data and code in order to reproduce the results. Finally, in Section 10 we draw conclusions and propose future work.

2. Related Work

Recent works have taken advantage of wearable sensors to perform simple activity recognition with them. Many of these works use motion sensors (e.g., accelerometer and gyroscope) to recognize simple physical activities like running, walking, sleeping, cycling, among other activities [19–24]. To perform simple activity recognition, usually, the accelerometer data is divided into fixed time windows, generally 2–10 s [19,23,25]. Then, time domain and/or frequency domain features [26] are extracted from the data. The set of features of the corresponding window is known as a feature vector or an instance. Those feature vectors are used to train and test classification models such as Decision Trees, Naïve Bayes, Support Vector Machines, and other types of classifiers [27].

There has also been research on complex and *long-term activities*. Martínez-Pérez *et al.*, implemented a system in a nursing home [1]. Their system integrates artifact behavior modeling, event interpretation and context extraction. They perform the inference based on how people handle different types of artifacts (devices to measure blood pressure, paper towel, cream, *etc.*). In their use case, they characterized the caregivers' activities of taking blood pressure, feeding, hygiene and medication of an elderly patient with restricted mobility over a period of ten days, achieving an accuracy of 91.35%. Gu *et al.* [28] focused on recognizing sequential, interleaved and concurrent activities like making coffee, ironing, drinking, using phone, watching TV, *etc.* They conducted their experiments in a smart home using sensors like accelerometers, temperature, humidity, light, *etc.* They also attached RFID tags to rooms, cups, teaspoons, and books. Their reported overall accuracy was 88.11%. One of the recent works in complex activity recognition is that of Cook *et al.* [29]. In their work, they do activity discovery and activity recognition with data collected from three smart apartments with several installed sensors and they deal with the interesting problem of having unlabeled activities. Each apartment housed an elderly during six months. They used infrared motion detectors and magnetic door sensors to recognize 11 different activities like bathing, cooking, eating, relaxing, *etc.* They reported accuracies of 71.08%, 59.76% and 84.89% for each of the three apartments.

There are also works that do complex activity recognition using wearable sensors. In the work of Huynh *et al.* [30], they used three wearable sensors boards: one on the right wrist, one on the right side of the hip and the last one on the right thigh. The sensor boards have 2D accelerometers and binary tilt switches. They recorded three different high level activities (housework, morning tasks and shopping) from one user giving a total of 621 min. They tested different algorithms and achieved the best accuracy (91.8%) when using histogram features with a Support Vector Machine. Mitchell *et al.*

used accelerometers from smartphones to classify sporting activities: five matches of soccer (15 players) and six hockey matches (17 players) [31]. They achieved a maximum F-measure accuracy of 87% by the fusion of different classifiers and extracting features using the Discrete Wavelet Transform. One of the advantages of using wearable sensors for activity recognition is that it is easy to uniquely identify the users. When using sensors installed in an environment with multiple residents, it becomes difficult to identify which user activated a specific sensor. Another advantage with wearable sensors is that the recognition can be performed almost in any place. A disadvantage of using a wearable sensor, e.g., just one accelerometer, is that it is not possible to detect activities that do not involve the movement of the part of the body that has the sensor. For example, if the accelerometer is in the user's pocket and he is sitting down, it may not be possible to tell whether he is working on a computer or maybe having dinner. Installed sensors and wearable sensors are complementary approaches to perform human activity recognition, both with strengths and weaknesses. For example, in [32] the authors used a multi-sensor approach with wearable (accelerometer) and installed sensors (cameras).

When recognizing simple activities, the approach of extracting features from window segments is appropriate since they last for only a few seconds. However, to recognize long-term activities from *accelerometer* data, generating fixed length windows is not suitable. For example, the long-term activity *working* could have small periods of the walking activity, but this does not mean that the user is commuting to another place. If a fixed time window method is used, it may be the case that the window includes only data about the walking activity so the model can confuse this activity with *commuting*. Generally, users perform long-term activities in a continuous manner and they do not have a clear separation boundary. In order to recognize more complex activities from accelerometer data, models that can deal with temporal information are needed. Examples of such models are Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), which have already been used extensively for activity recognition. For example, Lee and Cho used hierarchical HMMs to recognize activities like taking bus, shopping, walking, among other activities, using data collected with a smartphone [33]. Their method consists of first recognizing actions using HMMs and then using these actions sequences to feed higher level HMMs that perform the activity recognition. In [34], the authors used HMMs to recognize actions from several inertial sensors placed in different parts of the body. They used nine sensor boards in different parts of the body and their experiments included three subjects. In the previously mentioned work [29], Cook *et al.* also used a CRF and an HMM. On the other hand, Van Kasteren *et al.* used HMMs and CRFs to monitor activities for elderly care from data collected from a wireless sensor network in a smart home [35]. In the work of Vinh *et al.*, they used semi-Markov CRFs, which enabled them to model the duration and the interdependency of activities (dinner, commuting, lunch, office) [36]. Their reported precision was 88.47% on the dataset from [37], which consists of data collected during seven days using two triaxial accelerometers. Table 1 summarizes some of the related works for complex activity recognition.

Table 1. Related complex activity recognition works.

Work	# Activities	Sensors	Details
Martínez-Pérez *et al.* [1]	4: taking blood pressure, feeding, hygiene, medication	RFID, accelerometers, video cameras	91.35% accuracy, 1 patient during 10 days. 81 instances.
Gu *et al.* [28]	26: making coffee, ironing, using phone, washing clothes, *etc.*	accelerometers, temperature, humidity, light, RFID, *etc.*	Overall accuracy 88.11%, 4 subjects over a 4 weeks period. Collected instances 532.
Cook *et al.* [29]	11: bathing, cooking, sleeping, eating, relaxing, taking medicine, hygiene, *etc.*	infrared motion detectors and magnetic door sensors	Accuracies of 71.08%, 59.76% and 84.89% for each of the 3 apartments during a period of 6 months.
Huynh *et al.* [30]	3: housework, morning tasks and shopping.	2D accelerometers and tilt switches	Accuracy of 91.8% for 1 user and period of about 10 h.
Kasteren *et al.* [35]	bathing, dressing, toileting, *etc.*	reed switches, pressure mats, mercury contacts, passive infrared, float sensors and temperature sensors	4 different datasets
Tolstikov *et al.* [38]	7: leaving, toileting, showering, sleeping, breakfast, *etc.*	14 binary sensors	Maximum accuracy of 95.7% for 1 subject during 27 days.
Vinh *et al.* [36]	4: dinner, commuting, lunch and office work	2 triaxial accelerometers	Precision of 88.47% for data collected during 7 days.
Sung *et al.* [39]	12: cooking, talking on the phone, working on computer, *etc.*	Microsoft Kinect	Average precision 86.5%, data collected by 4 subjects
Gordon *et al.* [40]	7: drinking, gesticulating, put mug on table, meeting, presentation, coffee break, *etc.*	accelerometers attached to mugs	Average accuracy of 95% for single-user and maximum 96% for group activities. 3 subjects. In total over 45 mins. of collected data.

This work differs from the previous ones in the following aspects. (1) We use a single accelerometer in a wristwatch to perform long-term activity segmentation; (2) We propose a method to deal with the problem of intra-class fragmentation by finding possible subclasses of each class by means of clustering quality indices. We found that the overall recognition accuracy was improved (within our tested approaches) by incorporating prior knowledge such as the activities minimum lifespan and by finding sequence patterns that are common to each of the activities. *Subclassing* fragmented activities also helped to increase the overall recognition accuracy within our tested approaches.

3. Background

In this section we introduce the notations that will be used throughout this paper, as well as the background theory of Hidden Markov Models and Conditional Random Fields. From here on, we will use the notation introduced by Rabiner and Juang ([10], Chapter 6).

3.1. Hidden Markov Models

A Hidden Markov Model is a probabilistic graphical model consisting of a set of observations and states represented by random variables and can be defined as a 5-tuple $\langle N, M, A, B, \pi \rangle$ where:

- N is the number of states in the model indexed by $\{1, 2, ..., N\}$. The current state at time t is q_t.
- M is the number of distinct observation symbols denoted as $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M\}$.
- A is the state transition probability distribution. $A = \{a_{ij}\}$ where

$$a_{ij} = P(q_{t+1} = j \mid q_t = i), \quad 1 \leq i, j \leq N \tag{1}$$

  and $a_{ij} \geq 0 \quad \forall j, i; \sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
- B is the observation symbol probability distribution. $B = \{b_j(k)\}$ in which

$$b_j(k) = P(\mathbf{o}_t = \mathbf{v}_k \mid q_t = j), \quad 1 \leq k \leq M, \tag{2}$$

  defines the symbol distribution in state $j, j = 1, 2, ..., N$.
- π is the initial state distribution $\pi = \{\pi_i\}$ in which

$$\pi_i = P(q_1 = i), \quad 1 \leq i \leq N. \tag{3}$$

3.2. Conditional Random Fields

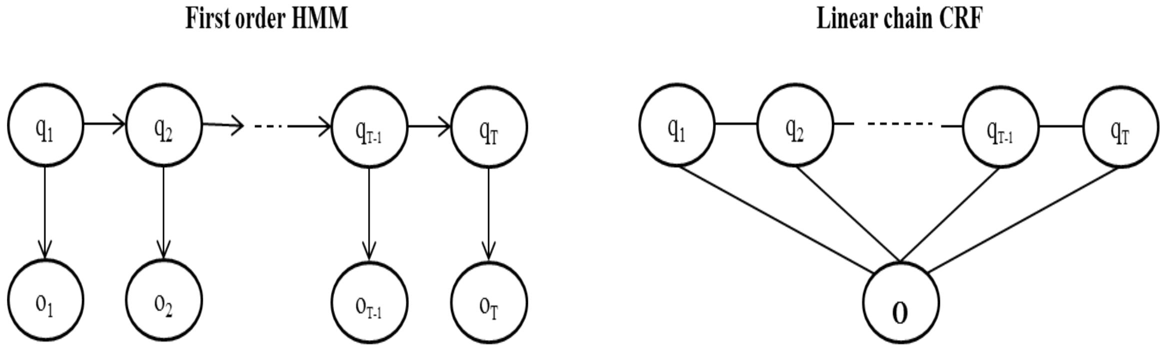
Hidden Markov Models are generative models that define a joint probability distribution $P(\mathbf{O}, \mathbf{Q})$ where $\mathbf{O}$ and $\mathbf{Q}$ are random variables. An observation $\mathbf{o}_t$ may only depend on the state at time t [41] (Figure 1). This is a strong assumption—in real world scenarios an observation may depend also on states at different times. For example, in part-of-speech tagging [42] the aim is to label a sequence of words with tags like "PERSON", "COMPANY", "VERB", *etc.* For example, if "*Martin Inc.* ...." is the sequence of words, as soon as we see "*Martin*" we may label it as "PERSON" but if we look at the next observation "*Inc.*" then we realize that the first tag should be "COMPANY". This type of dependencies can be easily modeled with a CRF.

Conditional Random Fields were first introduced by Lafferty *et al.* [15] to label sequenced data. Linear chain Conditional Random Fields define a conditional probability:

$$P(\mathbf{Q} \mid \mathbf{O}) = \frac{1}{Z} \exp\left(\sum_{t=1}^{T} \sum_{i=1}^{F} \omega_i f_i \left(q_{t-1}, q_t, O, t\right)\right) \tag{4}$$

where $\mathbf{Q}$ is the states sequence and $\mathbf{O}$ is the observations sequence. f_i and ω_i are feature functions and their respective weights and can take real values. F is the total number of feature functions.

Figure 1. A visual comparison between a first order HMM and a linear chain CRF. The HMM defines a joint probability $P(\mathbf{O}, \mathbf{Q})$ whereas the CRF defines a conditional probability $P(\mathbf{Q} \mid \mathbf{O})$. Note that an HMM only has access to the current observation $\mathbf{o}_t$ but the CRF has access to the entire observation sequence $\mathbf{O}$ at any given time.



Since the feature functions can take arbitrary real values, the Z term is used as a normalization factor to make it a valid probability and is defined as the sum of exponential number of sequences:

$$Z = \sum\nolimits_{\mathbf{Q}} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{F}\omega_i f_i\left(q_{t-1}, q_t, O, t\right)\right). \tag{5}$$

The feature functions take as arguments the previous and current state q_{t-1}, q_t, the entire observations sequence $\mathbf{O}$ and the current position t. Note that, unlike HMM that only have access to the current observation at any given time, CRF have access to the entire observations sequence. An example of a feature function may be:

$$f_1\left(q_{t-1}, q_t, O, t\right) = \begin{cases} 1 & \text{if } q_t = \text{COMPANY and} \\ & \quad \mathbf{O}_{t+1} = \text{Inc.} \\ 0 & otherwise. \end{cases} \tag{6}$$

This will have the effect of increasing the probability of state sequences that assign the tag COMPANY to an observation that is followed by the "*Inc.*" word (if the corresponding weight ω_1 is positive). In this case it returns a binary value but it could be any real value.

The main advantage of a discriminative model is that it can easily include overlapping features like in the COMPANY example. Interdependent features can also be included in generative models by enhancing the models or making simplifying independence assumptions. The first approach becomes difficult when trying to retain tractability. Making simplifying assumptions can be done and works well for some applications but may degrade the performance for some others [17]. For linear CRFs (which are the ones used in this work) inference algorithms can be performed efficiently by variants of the standard dynamic programming methods for HMMs. For general graphs, inference problems become intractable and approximate methods are needed. In general, computing the normalization factor Z requires exponential time; however, since for this application we want to know what is the most likely sequence of states regardless of the exact probability value, we can omit that computation.

To perform the activity inference, we used the Viterbi algorithm on both HMMs and CRFs. The constrained version of the Viterbi algorithm [16] allows to set the minimum number of consecutive states that can occur for each state i. This requires an additional parameter κ, which is an array that stores the minimum consecutive states for each activity:

$$\kappa(i) = n \in \mathbb{N}_{>0}, \quad 1 \leq i \leq N \tag{7}$$

Now we can take into account the minimum lifespan of each activity by coding this information in the κ array. This information was computed from the data by finding the minimum number of consecutive states of each activity and dividing them by 2 (to allow some deviance).

4. Data Collection

A GENEActiv [43] watch with an embedded triaxial accelerometer was used to collect the data. This watch can sample at up to 100 Hz. For our experiments, the sampling rate was set at 20 Hz because it has been shown that activity classification is high for sampling rates above or equal to 10 Hz but the accuracy decreases at a sampling rate of 5 Hz [44]. This watch also collects temperature and light intensity but they were not used. The watch was placed on the dominant hand of each user (Figure 2). The users performed their activities of daily living and took note of the start and end time of several of them. Data were collected by two subjects during 11 and 10 days, respectively, from approximately 9:00 to 23:00. The first subject was one of the researchers and the second subject was a volunteer that had no relation to this work. The first subject recorded 6 activities: (1) *Shopping*, which consists of picking and buying groceries at the supermarket; (2) *Showering*, which consists of taking a shower and dressing; (3) *Dinner*, which also takes into account breakfast and lunch time; (4) *Working*, which for most of the time is working with a computer but also involves taking rests and going to the bathroom; (5) *Commuting*, which includes by bus, by car, or walking; (6) *Brush Teeth*, which consists of both brushing the teeth and using a dental floss. Subject 2 tagged 4 activities: (1) *Commuting*; (2) *Lunch*, which also consists of breakfast and dinner time; (3) *Working time*, which for most of the time is office work; (4) *Exercise*, which consists of walking, running, stretching, *etc.* There is also an activity *not tagged* for data that was not labeled by the users. Figure 3 shows the raw accelerometer data and the tagged activities for one specific day of one of the subjects. Tables 2 and 3 show the number of instances and the duration of each activity for both subjects.

To reduce some noise, an average filter with a window length of 10 (Equation (8)) was used:

$$v_s(t) = \frac{1}{n} \sum_{i=t-n}^{t-1} v(i) \tag{8}$$

where v is the original vector, v_s is the smoothed vector and n is the window length.

Figure 2. Device used to collect the data.

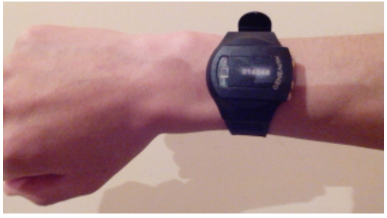


Figure 3. Raw accelerometer data and tagged activities during one specific day.

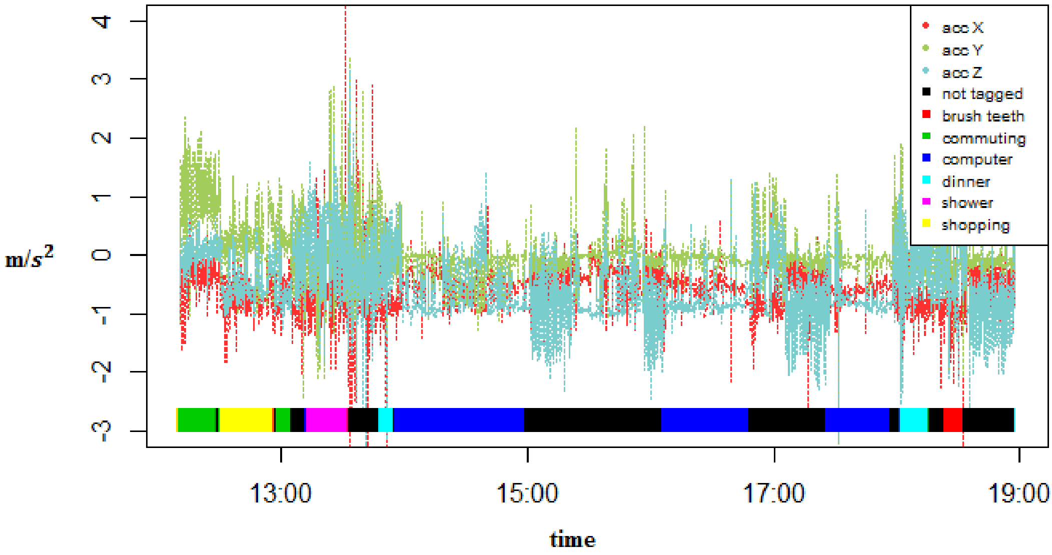


Table 2. Number of instances and duration of each self-reported activity (subject 1).

	Brush Teeth	Commuting	Computer	Dinner	Not Tagged	Shopping	Shower
# Instances	23	23	31	20	109	3	10
Total hours	2.9	9.4	32.4	5.0	50.1	1.0	3.5

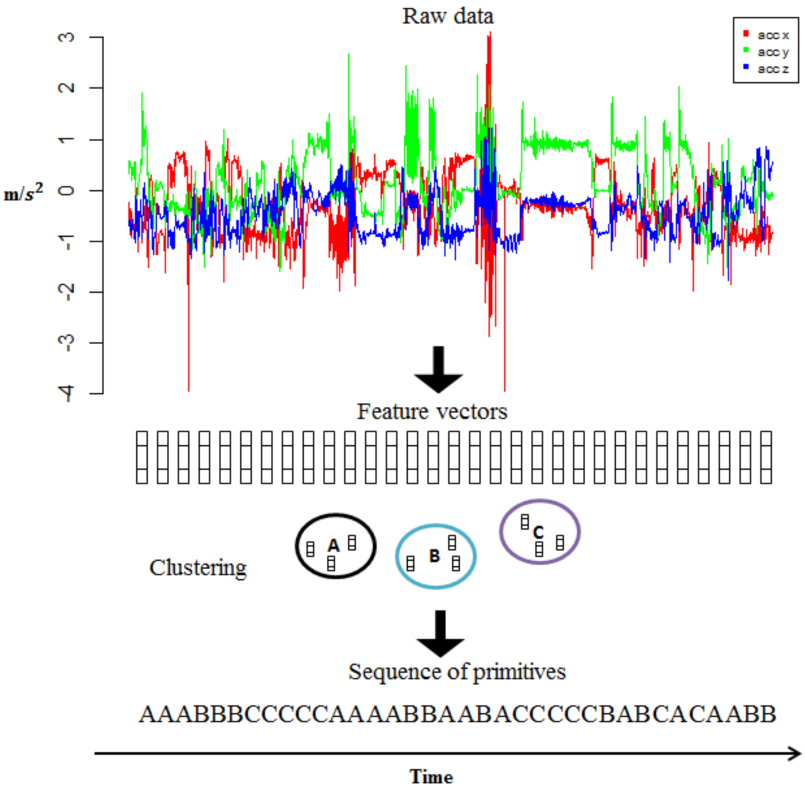
Table 3. Number of instances and duration of each self-reported activity (subject 2).

	Commuting	Lunch	Work	Exercise	Not Tagged
# Instances	54	17	21	3	95
Total hours	14.6	6.4	38.3	1.0	51.7

5. Sequence of Primitives Generation

A long-term activity is characterized by a sequence of primitives. These primitives are automatically discovered from the data. In the context of HMMs and CRFs they will be the observations. To obtain the sequence of primitives, the data is divided into windows of fixed length l (we used $l = 60$, which is ≈ 3 s) with an overlap of 33%, because in [25] it was shown that small window sizes lead to better results than using longer window sizes. For each window segment, the next set of 12 features were computed: mean and variance for each of the *x,y* and *z* axes; the correlation between every pair of axes; and the mean, variance and average derivative of the magnitude. The resulting feature vectors do not have a label or class. The task consists of categorizing the feature vectors into different groups such that instances that are very similar to each other belong to the same group. For this step we used the k-means clustering method, where k represents the number of clusters we want. Next, the center or the centroid of each cluster will correspond to a primitive with a unique ID. This clustering step is just performed on the training set. To assign a label/class to an instance, we get the ID of the closest centroid and assign it as the label of the instance. Figure 4 depicts the overall process of vector quantization.

Figure 4. The acceleration signals are transformed into sequences of primitives using vector quantization.



6. CRF Feature Functions and Training

In this section we describe each of the three feature functions that were included in the model and how they were estimated from the training data.

6.1. f_1 Observation Symbol Probability Distribution

This feature function is equivalent to Equation (2) for HMMs and defines the probability of observing an specific symbol (*primitive*) given some state (activity).

$$f_1(q_{t-1}, q_t, \mathbf{O}, t) = \log\left(P(\mathbf{o}_t \mid q_t)\right). \tag{9}$$

These probabilities are estimated from the training data as:

$$P(\mathbf{o}_k \mid q) = \frac{\text{number of primitives of class } k \text{ in } q}{\text{total number of primitives in } q}, {}_{\substack{1 \leq q \leq N \\ 1 \leq k \leq M}}$$

The reason for the $\log()$ function is to make this feature equivalent to the symbol probability distribution of an HMM.

6.2. f_2 State Transition Probability Distribution

This is equivalent to the state transition probability distribution of an HMM (Equation 1).

$$f_2(q_{t-1}, q_t, \mathbf{O}, t) = \begin{cases} \log\left(\pi_{q_t}\right) & \text{if } t = 1 \\ \log\left(P(q_t \mid q_{t-1})\right) & \text{otherwise} \end{cases} \tag{10}$$

where π_i are the initial probabilities that were set to be uniform. The probability of transitioning from one state to another $P(q_t \mid q_{t-1})$ is estimated by computing the transition frequency for every pair of states. Generally, the probability of transitioning from a state to itself will be much higher than transitioning to another state.

6.3. f_3 Sequence Patterns (k-Mers)

The purpose of this function is to include information about the sequence structure, *i.e.*, to find overlapping sequences of primitives that occur together. In bioinformatics a *k-mer* (also called *n*-grams, *n*-mers, *l*-tuples) ([45], p. 308) is a string of length k and $Count(Text, Pattern)$ is the number of times a *k-mer* $Pattern$ appears as a substring of $Text$. For example:

$Count($ACA**CTAC**TGCATA**CTACTAC**CT, **CTAC**$) = 3$ Note that the patterns can be overlapped. Usually, *k-mers* are used in DNA sequence analysis [45]. In text processing they are usually called *n*-grams [46]. The total number of different *k-mers* is M^k; recall that M is the number of distinct observation symbols. The feature function f_3 is defined as:

$$f_3(q_{t-1}, q_t, \mathbf{O}, t) = \log\left(P(\mathbf{o}_t, ..., \mathbf{o}_{t+k-1} \mid q_t)\right) \tag{11}$$

which is the probability of finding a specific *k-mer* given an activity. This feature function is based on one of the implications of Zipf's law and is described in [46] in the context of document classification as: "if we are comparing documents from the same category they should have similar N-gram frequency distributions". Note that this feature function is similar to f_2 when $k = 2$ but instead of activity transitions, it defines *primitives* transitions, *i.e.*, the probability of some primitive v_i occurring after some primitive v_j given an activity. In the training phase we estimate these probabilities as:

$$P(\mathbf{o}_t, ..., \mathbf{o}_{t+k-1} \mid q) = \frac{Count(PrimSeq_i, k\text{-}mer)}{\mid PrimSeq_i \mid -k + 1}, {}_{1 \leq i \leq N}$$

for all M^k possible *k-mers*. $PrimSeq_i$ is the concatenation of the *primitives* sequence for all long-term activities of type i from the training set.

7. Subclassing Fragmented Classes

One of the problems when tagging the activities in the training phase is the intra-class fragmentation. This means that several observations may belong to the same class even though they are completely different. For example, a user may tag an activity as *commuting* but sometimes he may commute mainly by bus and other times by walking, which are very different from the acceleration point of view. This will produce noise in the models since the training will take into account all observations of the same class in conjunction, but in the recognition phase an observation may belong to just one subclass (commuting by bus or walking). Of course the user could create two different classes at training time (bus commute and walk commute) but this would be a tedious task and the subclasses may not be obvious. What we propose is the following sequence of steps:

(1) For each class C_i in the training set, find if it is fragmented.
(2) If it is fragmented, assign each of its observations to their corresponding subclass $C_{i,j}$ for $j = 1...\text{F}$ where F is the total number of fragments.
(3) If it is not fragmented, assign all observations to a single subclass $C_{i,1}$.
(4) Build/train the models with the subclasses as if they were independent classes.
(5) Predict the subclass of the observations in the test set.
(6) For the test set, replace each predicted subclass $C_{i,j}$ with the original class C_i.
(7) Validate the results with the ground truth (original class labels).

Figure 5 shows the overall process. One way to find if a class is fragmented is by visual inspection. One approach to perform the visual inspection is to cluster the observations using a method that does not require specifying the number of groups a priori, e.g., using hierarchical clustering and then plotting the resulting dendrogram. Figure 6 shows the resulting dendrogram when applying hierarchical clustering to the observations of subject 1. From this dendrogram, we can see that the *working with computer* class is fragmented roughly into two, the *shopping* activity looks uniform, the *commuting* activity looks like it is fragmented into two, *etc.* In some cases it is not obvious in how many fragments a class is divided into. For example, the *brush teeth* activity could be divided into three or maybe four subclasses so this approach is subjective and may just work well when the number of classes is small and the separation of the fragments is more or less clear.

To perform the clustering, each long-term activity is represented as a sequence of *primitives* and will be characterized with a feature vector of size M, where M is the total number of different *primitives*, *i.e.*, the alphabet size. The feature vector stores the frequencies of each of the *primitives* so each long-term activity is characterized by a distribution of *primitives* (a *histogram*).

The visual approach for finding the fragments of a class is feasible when (1) we deal with a small set of activities, and (2) when we have predefined knowledge about the subclasses, *i.e.*, an expert is able to validate from human experience. In the case of external validity, that is, when the user knows the class labels or the number of subclasses, good cluster structure is accomplished when the predicted classes are the same as the predefined labels of the dataset. When either of the two characteristics mentioned before is not fulfilled, a more flexible approach must be used. In the absence of prior knowledge, the use of relative validity metrics [47] may offer a good approximation to the real segmentation.

Figure 5. Overall process for performing the subclassing.

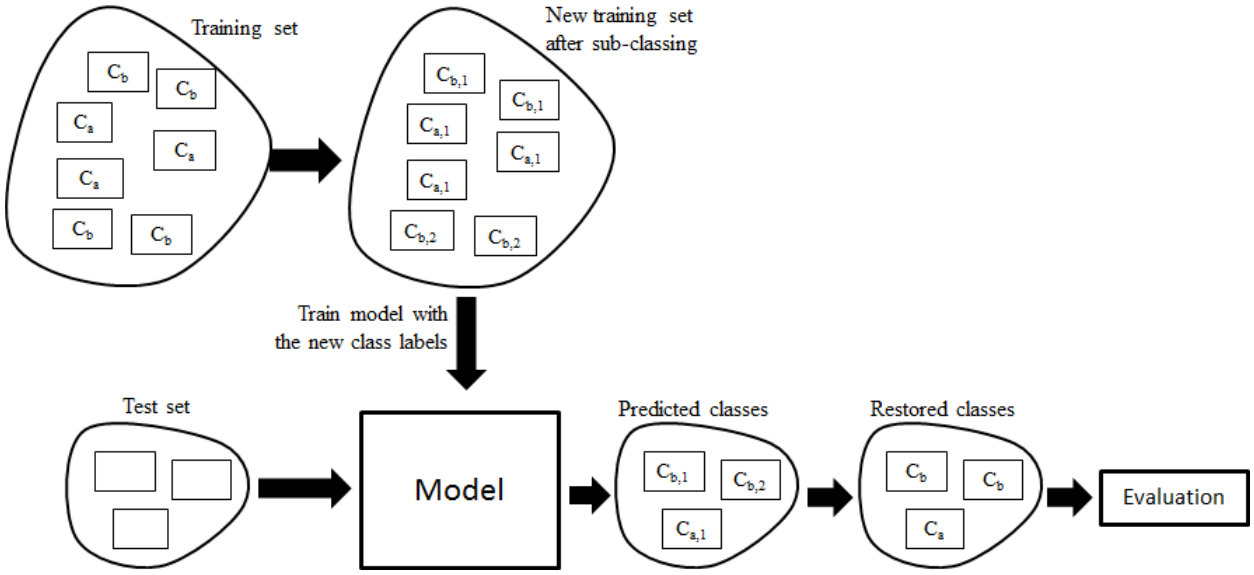
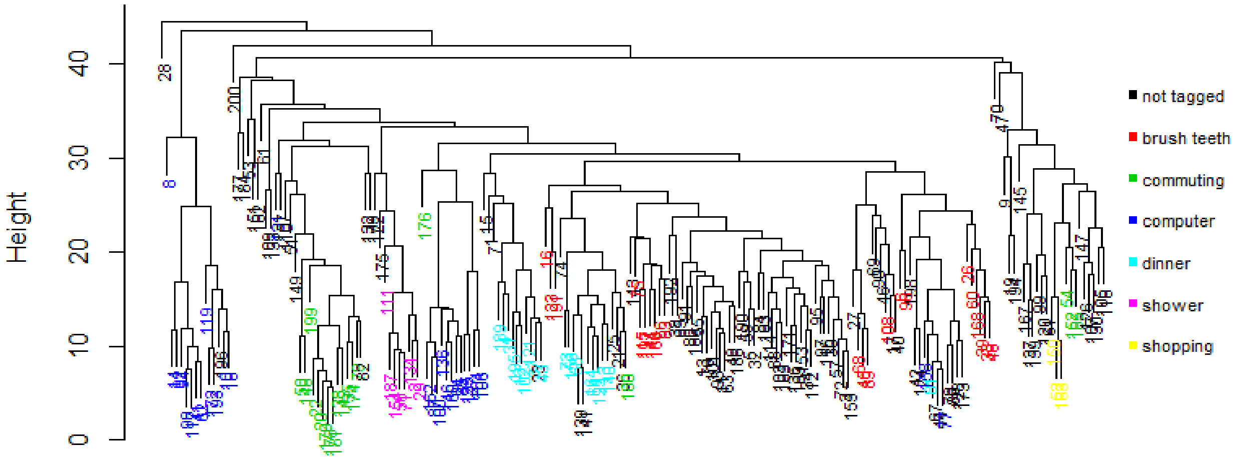


Figure 6. Resulting dendrogram of hierarchical clustering for subject 1.



Let us recall that the objective of partitioning techniques is to cluster groups of elements that are highly related in terms of similarity. As a consequence, the idea is to find clusters where the elements of each group are as close to each other as possible (*i.e.*, compactness), and the distance between the clusters is widely spaced (*i.e.*, separation). In this case, cluster structure is measured in terms of clustering quality indices [18], which indicate the quality of cluster structure based on compactness and separation. Some relative validity indices widely used include Dunn Index, Davies-Bouldin Index, Silhouette Index, Maulik-Bandoypadhyay Index, PBM Index [47–49].

In this work we used the Silhouette [50], PBM [49] and Generalized Dunn's Index (GDI33) [51] clustering quality indices. Silhouette and GDI33 were chosen because in [18] they proved to work well with different datasets. PBM was chosen because it is more recent and in the original paper [49] it was claimed that the results were superior than other previously proposed indices.

The following process to get the optimal number of fragments with respect to some quality measure is based on the steps described in [52] and was applied just to the training set:

(1) Cross-validate the training set.
(2) For each class C_i, cluster the observations that belong to C_i into k groups, for $k = 2...maxK$. $maxK$ is a predefined upper bound on the allowed number of fragments. Compute the quality index of the clustering result for each k.
(3) Subclass C_i into k subclasses where k is the number of groups that yielded the optimal quality index.
(4) Cross-validate the training set with the new subclasses.
(5) If the resulting accuracy improves the result of step 1, keep the new subclassing for C_i.

For our experiments, in the cross-validation step, instead of building the HMMs and CRFs, we used the Naive Bayes classifier with the activities *histograms* as input. The reason of validating the subclassing step with a simpler classifier is because it is much faster than building the HMMs and CRF models and this step has to be done $maxK$ times for each class. For the clustering step, we used the *k*-means algorithm. One of the advantages of finding the fragments using a quality index is that it can be done automatically without user intervention as opposed to the visual approach described earlier. It is worth noting that the discovered sub-activities may not correspond to "real activities" since the clustering process is based in kinematic features and, as a consequence, activities are split according to their qualities rather than their real sub-type. However, for the purposes of this work this is not relevant since the subclassing is exploited to increase the overall accuracy rather than used to understand what these sub-activities are.

8. Experiments and Results

Recall that a long-term activity is represented as a sequence of *primitives*. A *run* will be defined as a concatenation of long-term activities that occur consecutively, *i.e.*, an entire day. We performed five experiments: (1) without subclassing; (2) with fixed subclassing; (3) with silhouette subclassing (4) with PBM subclassing; (5) with GDI33 subclassing. Fixed subclassing means that the number of subclasses for each class were set manually by visual inspection of the dendrogram. In each of the experiments we compared three different methods to perform the activity segmentation:

(1) **HMM**: Viterbi algorithm on an HMM without constraint.
(2) **HMM + C**: Viterbi algorithm on an HMM with the k-minimum-consecutive-states constraint.
(3) **CRF**: Viterbi algorithm on a CRF with the k-minimum-consecutive-states constraint and adding information about sequence patterns.

Since long-term activities are user-dependent, *i.e.*, for one person, the *working* activity may involve doing office work and for another person the *working* activity may be more physically demanding, individual models with the same parameters were built for each subject. Leave-one-day-out cross validation was used to validate the results. This means that for each day i, all other days j, $j \neq i$ are used as training set and day i is used as the test set. The accuracy was computed as the percentage of correct state predictions. Due to computational limitations, the k for *k-mers* was set to be a small number (in this case, 2) because the number of *k-mers* grows exponentially ($M^k = 300^2 = 90,000$ possible different patterns). The number of *primitives* was set to 300 because the accuracy does not have a significant increase for values greater than 250 (Figure 7). The number of primitives parameter depends on the sensor data and the features extracted. In our previous work [9] we collected data from a cellphone located on the user's belt and used a different set of features. In that case, a parameter value greater or equal to 15 gave the best results. In [11] they used a MotionNode sensor to detect simple activities. In that case, an alphabet size of around 150 gave the best results. The number of primitives will depend on each application and is determined empirically as stated in [11]. The overall results for each of the five experiments are listed in Table 4.

Figure 7. Accuracy as number of primitives increase.

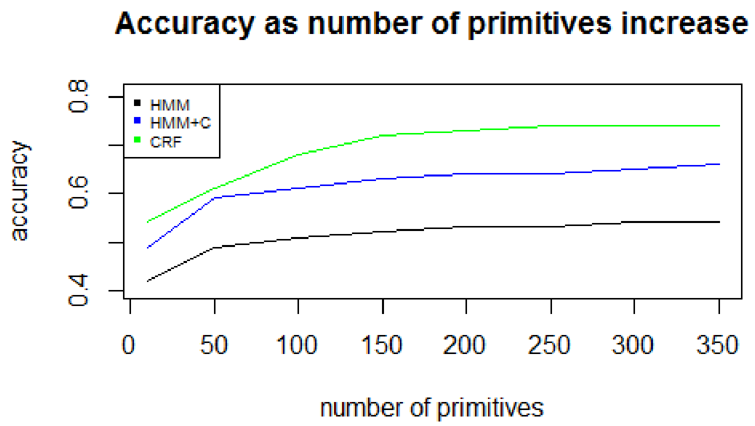


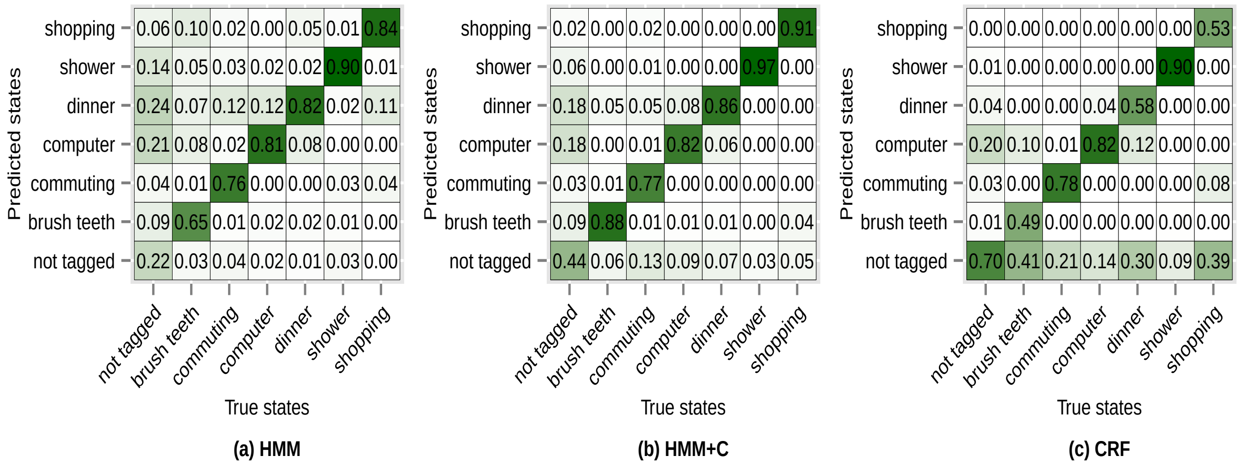
Table 4. Overall accuracies for the five experiments for the two subjects.

	HMM	HMM + C	CRF
Experiment 1: no subclassing	60.7%	68.8%	75.1%
Experiment 2: fixed	64.7%	71.9%	77.1%
Experiment 3: silhouette	64.1%	71.0%	76.3%
Experiment 4: PBM	63.9%	70.8%	76.2%
Experiment 5: GDI33	64.1%	71.0%	76.3%

From Table 4 it can be seen that for the five experiments the accuracy increases as more information is added to the model—first adding the minimum activities lifespan (HMM + C) and then adding both the minimum lifespan and the sequence patterns (CRF). To assess the influence of the transitions between activities, instead of learning the transition matrix from the data we set it to have a uniform distribution between any two possible transitions (to represent a lack of information). In this case, the overall accuracy (no subclassing) was 52.0%, 67.9%, and 75.5% for HMM, HMM + C and CRF, respectively. This suggests that the transitions between activities had a considerable influence in the first model (HMM), but HMM + C and CRF were more robust to the choice of the transition matrix. The type of activities considered in this work may have strong predictable diurnal patterns. As a baseline, we performed the activity recognition with no accelerometer data by just using information of the time of the day. First, the predicted activity at a given time was set as the activity with highest probability of occurring at that specific time of the day based on the training data. This resulted in an overall accuracy of 44%. Then, we used the Viterbi algorithm on an HMM with the time of the day as observations, resulting in an overall accuracy of 47%. These results suggest that just using accelerometer data, long-term activities can be detected independently of the time of the day and with higher accuracy than relying on just time daily patterns. This is especially true for *high-entropy subjects*, which tend to be more variable and do not follow a fixed schedule and are thus harder to predict based on their daily routines patterns [53].

Figure 8 shows the confusion matrix of experiment 1 for subject 1. From Figure 8a we can see that 76% of the observations that were generated by the *commuting* activity were correctly classified as being generated by the *commuting* activity. We can also see that 2% of the observations that were generated by the *commuting* activity were misclassified as if they had been generated by the *shopping* activity. The antidiagonal of the matrices shows the sensitivity (true positive rate). For example *brush teeth* has a sensitivity of 0.65, commuting has a sensitivity of 0.76, *etc.* One thing to note is that the sensitivity tends to increase for all activities when using HMM + C with respect to HMM. When using a CRF, the sensitivity for the *not tagged* states increases significantly but for some of the activities it drastically decreases. Even though the overall accuracy when using a CRF increases, the sensitivity of some activities decreases. It seems that using HMM + C is a good choice for a tradeoff between overall accuracy and activity sensitivity. One of the reasons that increasing the sensitivity of *not tagged* states and decreasing the sensitivity of the other activities yield an overall higher accuracy is because most of the states are of type *not tagged* (Tables 5 and 6). Almost half of the data was not tagged by the users. Tagging the activities is tedious for a user and consumes time. One possible approach that we are going to explore for future work is to fill missing tags using information from other users. The idea is not to force the user to continuously tag the data but to let the system automatically fill missing information obtained from other sources (a crowdsourcing approach).

Figure 8. Confusion matrices for experiment 1 (subject 1): no subclassing.



(a) HMM

Predicted states	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.06	0.10	0.02	0.00	0.05	0.01	0.84
shower	0.14	0.05	0.03	0.02	0.02	0.90	0.01
dinner	0.24	0.07	0.12	0.12	0.82	0.02	0.11
computer	0.21	0.08	0.02	0.81	0.08	0.00	0.00
commuting	0.04	0.01	0.76	0.00	0.00	0.03	0.04
brush teeth	0.09	0.65	0.01	0.02	0.02	0.01	0.00
not tagged	0.22	0.03	0.04	0.02	0.01	0.03	0.00

(b) HMM+C

Predicted states	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.02	0.00	0.02	0.00	0.00	0.00	0.91
shower	0.06	0.00	0.01	0.00	0.00	0.97	0.00
dinner	0.18	0.05	0.05	0.08	0.86	0.00	0.00
computer	0.18	0.00	0.01	0.82	0.06	0.00	0.00
commuting	0.03	0.01	0.77	0.00	0.00	0.00	0.00
brush teeth	0.09	0.88	0.01	0.01	0.01	0.00	0.04
not tagged	0.44	0.06	0.13	0.09	0.07	0.03	0.05

(c) CRF

Predicted states	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.00	0.00	0.00	0.00	0.00	0.00	0.53
shower	0.01	0.00	0.00	0.00	0.00	0.90	0.00
dinner	0.04	0.00	0.00	0.04	0.58	0.00	0.00
computer	0.20	0.10	0.01	0.82	0.12	0.00	0.00
commuting	0.03	0.00	0.78	0.00	0.00	0.00	0.08
brush teeth	0.01	0.49	0.00	0.00	0.00	0.00	0.00
not tagged	0.70	0.41	0.21	0.14	0.30	0.09	0.39

We used a paired Student's t-test to statistically validate whether or not there is a significant difference when performing the recognition without subclassing and with silhouette subclassing. This was done for each of the three algorithms: HMM, HMM + C and CRF. The null hypothesis is that $\mu_0 - \mu_s = 0$ (the mean accuracy when not subclassing μ_0 is the same as when using silhouette subclassing μ_s). The alternative hypothesis is that $\mu_s - \mu_0 > 0$, *i.e.*, there is a significant increase in the overall accuracy when using silhouette subclassing. The significance level for the tests was set at $\alpha = 0.05$.

Table 5. Percent of states of each class (subject 1).

	Brush Teeth	**Commuting**	**Computer**	**Dinner**	**Not Tagged**	**Shopping**	**Shower**
(%)	2.8	9.0	30.9	4.8	47.9	1.0	3.4

Table 6. Percent of states of each class (subject 2).

	Commuting	**Exercise**	**Lunch**	**Not Tagged**	**Work**
(%)	13.1	0.9	5.8	46.1	33.8

Figure 9 shows the paired box plots that represent the accuracy for each of the 21 days. It can be seen how the mean accuracy increases when performing the silhouette subclassing for the three algorithms. Table 7 shows the results of the formal statistical tests to validate this. The second column presents the difference of the overall accuracy when not subclassing and when using silhouette subclassing. The *t*-test assumes that the data is normally distributed, so we used the Shapiro–Wilk normality test [54] to validate this assumption and the resulting *p*-values are shown in the third column. The fourth column shows the resulting *p*-value of the paired *t*-test. The null hypothesis of the Shapiro–Wilk test is that the data has normal distribution. Since the *p*-value for HMM and HMM + C is smaller than the significance value 0.05, we can conclude that the data is not normal, which violates the assumption. The *t*-test for this case can still be done but the results should be taken with caution. The *t*-test *p*-values for the three algorithms are smaller than the significance level 0.05, so we can conclude that there is a significant increase in the accuracy when using silhouette. Since the normal assumption was violated for the CRF case, we also validated the results with a Mann–Whitney U test, which is non-parametric and does not assume a normal distribution of the variables. The results of these tests are shown in the last column and all *p*-values also resulted significant.

Figure 9. Paired box plots showing the accuracies for the 21 days.

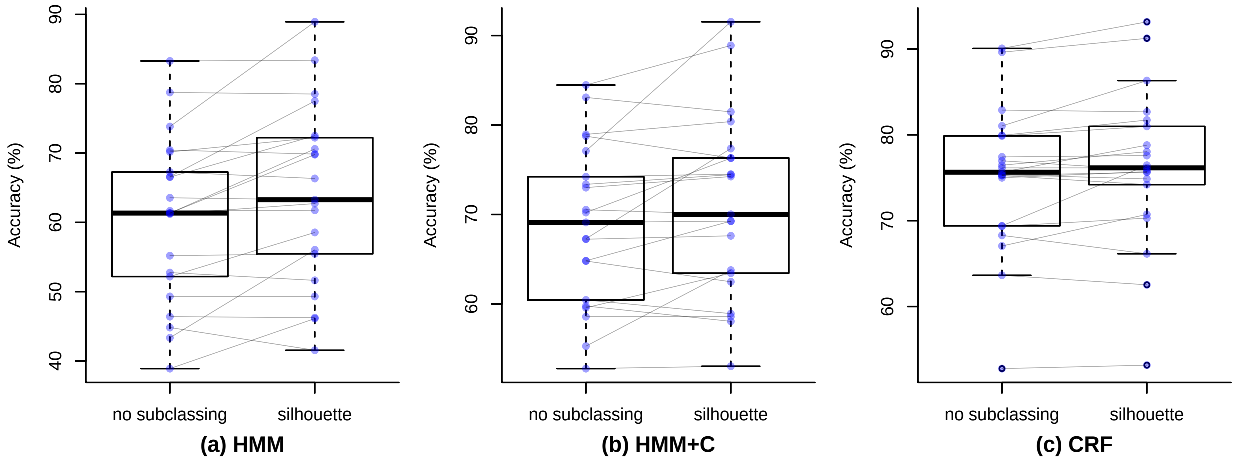
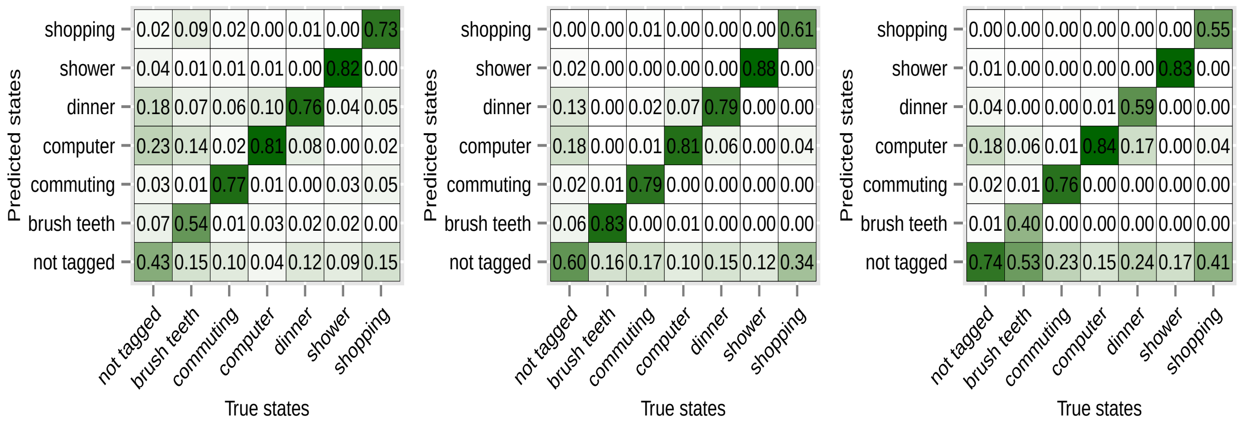


Table 7. Resulting p-values of the statistical tests (μ_0: mean accuracy with no subclassing, μ_s: mean accuracy with silhouette subclassing).

Algorithm	$\mu_0 - \mu_s$	**Shapiro–Wilk *p*-Value**	***t*-Test *p*-Value**	**Mann–Whitney *p*-Value**
HMM	-3.5	$p << 0.05$	$p << 0.05$	$p << 0.05$
HMM + C	-2.2	$p << 0.05$	$p << 0.05$	$p << 0.05$
CRF	-1.1	$p >> 0.05$	$p << 0.05$	$p << 0.05$

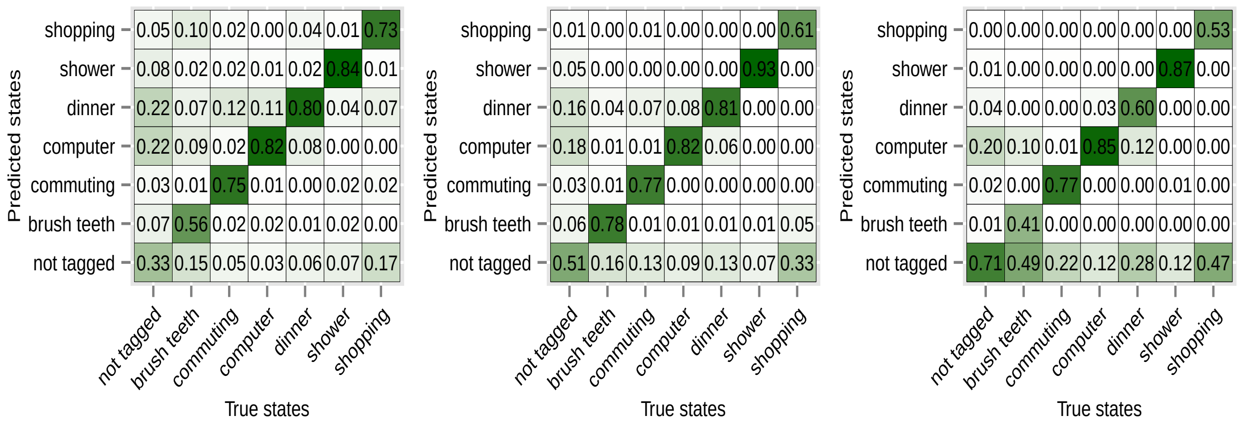
The results of using other clustering quality indices (PBM, GDI33) produced very similar confusion matrices and overall accuracies compared with using the silhouette index. Higher accuracies were obtained when applying a fixed subclassing by visually inspecting the resulting dendrogram, but the difference is small compared with the use of clustering indices. Figures 10–13 show the confusion matrices for experiments 2,3,4 and 5 for subject 1.

Figure 10. Confusion matrices for experiment 2 (subject 1): fixed subclassing.



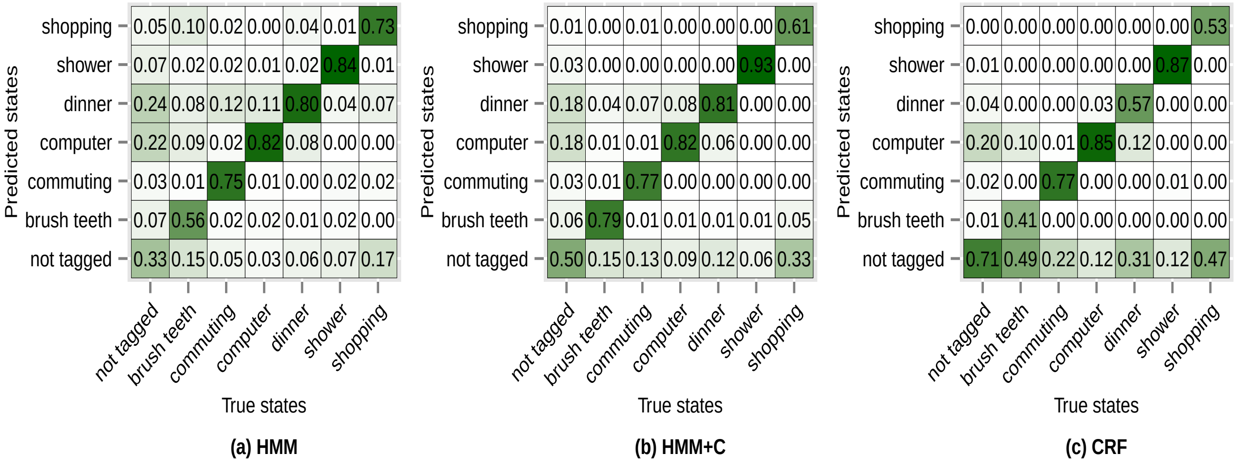
(a) HMM **(b) HMM+C** **(c) CRF**

Figure 11. Confusion matrices for experiment 3 (subject 1): silhouette subclassing.



(a) HMM **(b) HMM+C** **(c) CRF**

Figure 12. Confusion matrices for experiment 4 (subject 1): PBM subclassing.



(a) HMM

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.05	0.10	0.02	0.00	0.04	0.01	0.73
shower	0.07	0.02	0.02	0.01	0.02	0.84	0.01
dinner	0.24	0.08	0.12	0.11	0.80	0.04	0.07
computer	0.22	0.09	0.02	0.82	0.08	0.00	0.00
commuting	0.03	0.01	0.75	0.01	0.00	0.02	0.02
brush teeth	0.07	0.56	0.02	0.02	0.01	0.02	0.00
not tagged	0.33	0.15	0.05	0.03	0.06	0.07	0.17

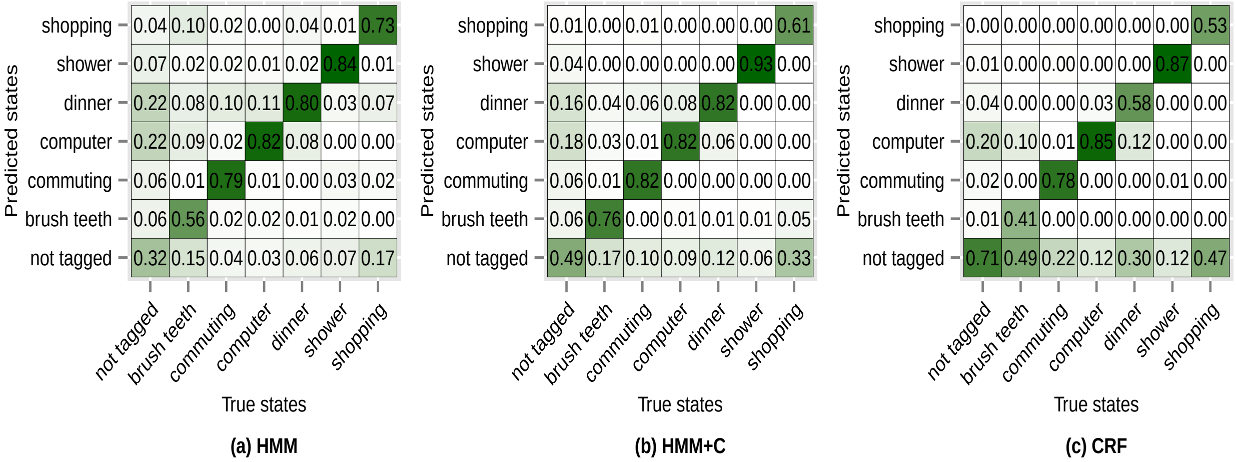
(b) HMM+C

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.01	0.00	0.01	0.00	0.00	0.00	0.61
shower	0.03	0.00	0.00	0.00	0.00	0.93	0.00
dinner	0.18	0.04	0.07	0.08	0.81	0.00	0.00
computer	0.18	0.01	0.01	0.82	0.06	0.00	0.00
commuting	0.03	0.01	0.77	0.00	0.00	0.00	0.00
brush teeth	0.06	0.79	0.01	0.01	0.01	0.01	0.05
not tagged	0.50	0.15	0.13	0.09	0.12	0.06	0.33

(c) CRF

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.00	0.00	0.00	0.00	0.00	0.00	0.53
shower	0.01	0.00	0.00	0.00	0.00	0.87	0.00
dinner	0.04	0.00	0.00	0.03	0.57	0.00	0.00
computer	0.20	0.10	0.01	0.85	0.12	0.00	0.00
commuting	0.02	0.00	0.77	0.00	0.00	0.01	0.00
brush teeth	0.01	0.41	0.00	0.00	0.00	0.00	0.00
not tagged	0.71	0.49	0.22	0.12	0.31	0.12	0.47

Figure 13. Confusion matrices for experiment 5 (subject 1): GDI33 subclassing.



(a) HMM

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.04	0.10	0.02	0.00	0.04	0.01	0.73
shower	0.07	0.02	0.02	0.01	0.02	0.84	0.01
dinner	0.22	0.08	0.10	0.11	0.80	0.03	0.07
computer	0.22	0.09	0.02	0.82	0.08	0.00	0.00
commuting	0.06	0.01	0.79	0.01	0.00	0.03	0.02
brush teeth	0.06	0.56	0.02	0.02	0.01	0.02	0.00
not tagged	0.32	0.15	0.04	0.03	0.06	0.07	0.17

(b) HMM+C

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.01	0.00	0.01	0.00	0.00	0.00	0.61
shower	0.04	0.00	0.00	0.00	0.00	0.93	0.00
dinner	0.16	0.04	0.06	0.08	0.82	0.00	0.00
computer	0.18	0.03	0.01	0.82	0.06	0.00	0.00
commuting	0.06	0.01	0.82	0.00	0.00	0.00	0.00
brush teeth	0.06	0.76	0.00	0.01	0.01	0.01	0.05
not tagged	0.49	0.17	0.10	0.09	0.12	0.06	0.33

(c) CRF

	not tagged	brush teeth	commuting	computer	dinner	shower	shopping
shopping	0.00	0.00	0.00	0.00	0.00	0.00	0.53
shower	0.01	0.00	0.00	0.00	0.00	0.87	0.00
dinner	0.04	0.00	0.00	0.03	0.58	0.00	0.00
computer	0.20	0.10	0.01	0.85	0.12	0.00	0.00
commuting	0.02	0.00	0.78	0.00	0.00	0.01	0.00
brush teeth	0.01	0.41	0.00	0.00	0.00	0.00	0.00
not tagged	0.71	0.49	0.22	0.12	0.30	0.12	0.47

9. Reproducibility

In order to make the results reproducible, we made the data and code available for download [55,56]. The implementation was coded in R and Java programming languages. The code uses the HMM R package [57] and a modified version of its Viterbi function in order to include the k-minimum-consecutive states constraint. Both the data and the code files include a description of the data and the instructions to run the code.

10. Conclusions

In this work, we performed long-term activity segmentation from accelerometer data collected with a wristwatch. The long-term activities were transformed into sequences of simple activities by using vector quantization. HMMs and CRFs were used to perform the segmentation. It was shown how adding additional information to the models helped to increase the overall accuracy of the tested approaches. The additional information consisted of the minimum lifespan of each of the activities and sequence patterns (*k-mers*). Most of the works on complex activity recognition use a fixed infrastructure of sensors. There are also works that perform the recognition using different types of wearable sensors as described in Section 2. In this work, we explored the use of a single accelerometer to segment different long-term activities, which may be used as an indicator of how independent a person is and as a source of information to healthcare intervention applications. Most of the works also assume that a complex activity has always the same distribution of simple activities that compose it, which may not be the case since the same conceptual activity can be performed in very different ways. To deal with this issue, we introduced the idea of subclassing and described how to reduce the impact of intra-class fragmentation by subclassing the activities using visual inspection and clustering quality indices. The most similar works are [30,36], which achieved an accuracy of 91.8% and a precision of 88.47%, respectively. The former collected 10 h of data from one user and the latter used a dataset consisting of 7 days of data from one user. In this work, we achieved accuracies between 70.0% and 77% when using subclassing with HMM + C and CRFs for two different users (total 21 days of data) and 7 different activities using just one sensor. One of the limitations is that both users reported similar activities and in both cases the *working* activity involved office work, thus further evaluation with a wider range of users and activities is still needed to test the generalization of the method to other possible scenarios and types of users. Another limitation is that the difference between simple and long-term activities was explained to the volunteer subject before the data collection process, hence only long-term activities were tagged. As a consequence, the system assumes that the tagged data consist of just long-term activities but in real world scenarios the users should be allowed to personalize an activity recognition system with any type of activity regardless of its type. In addition, further evaluation needs to be done to see how well this approach could be applied to detecting activities in real time rather than doing the computations offline. This is important in order to provide direct support to individuals while the activity is taking place rather than waiting after the data has been collected.

In our experiments, most of the data was not labeled by the user, which can cause several problems. For example, the user may not have labeled a known activity i but at test time it may be correctly classified as i. Since it does not initially have a label, it will be marked as misclassified. To overcome this, for future work we will use a crowdsourcing approach to complete missing information. The idea of using crowdsourcing for activity recognition from video data is already being explored [58,59]. However, for accelerometer data it presents several challenges because it is hard to classify an activity based on visual inspection.

Acknowledgments

Enrique would like to thank Juan P. García-Vázquez for his suggestions and comments and Consejo Nacional de Ciencia y Tecnología (CONACYT) and the AAAmI research group at Tecnológico de Monterrey for the financial support in his Ph.D. studies.

Author Contributions

Enrique Garcia-Ceja is the main author of this research work and manuscript, with Ramon Brena as advisor. The main contribution of Jose C. Carrasco-Jimenez was in Section 7 of this work but also in the

revision of each of the other sections. Leonardo Garrido contributed to the revision of the overall work and manuscript and provided insightful comments and suggestions.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Martínez-Pérez, F.E.; González-Fraga, J.A.; Cuevas-Tello, J.C.; Rodríguez, M.D. Activity Inference for Ambient Intelligence through Handling Artifacts in a Healthcare Environment. *Sensors* **2012**, *12*, 1072–1099.
2. Han, Y.; Han, M.; Lee, S.; Sarkar, A.M.J.; Lee, Y.K. A Framework for Supervising Lifestyle Diseases Using Long-Term Activity Monitoring. *Sensors* **2012**, *12*, 5363–5379.
3. Jawbone UP. Available online: https://jawbone.com/up (accessed on 24 November 2014).
4. Pebble. Available online: https://getpebble.com/ (accessed on 24 November 2014).
5. Lawton, M.; Brody, E. Instrumental Activities of Daily Living Scale (IADL). Available online: http://ciir.cs.umass.edu/ dfisher/cs320/tablet/Surveys.pdf (accessed on 24 November 2014).
6. Lane, N.D.; Mohammod, M.; Lin, M.; Yang, X.; Lu, H.; Ali, S.; Doryab, A.; Berke, E.; Choudhury, T.; Campbell, A. Bewell: A smartphone application to monitor, model and promote wellbeing. In Proceedings of the 5th International ICST Conference on Pervasive Computing Technologies for Healthcare, Dublin, Ireland, 23–26 May 2011; pp. 23–26.
7. Aung, M.; Bianchi-Berthouze, N.; Watson, P.; Williams, A.D.C. Automatic Recognition of Fear-Avoidance Behavior in Chronic Pain Physical Rehabilitation. In Proceedings of the International Conference on Pervasive Computing Technologies for Healthcare, Oldenburg, Germany, 20–23 May 2014.
8. Singh, A.; Klapper, A.; Jia, J.; Fidalgo, A.; Tajadura-Jiménez, A.; Kanakam, N.; Bianchi-Berthouze, N.; Williams, A. Motivating People with Chronic Pain to Do Physical Activity: Opportunities for Technology Design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14), Toronto, ON, Canada, 21 May 2014.
9. Garcia-Ceja, E.; Brena, R. Long-Term Activity Recognition from Accelerometer Data. *Procedia Tech.* **2013**, *7*, 248–256.
10. Rabiner, L.; Juang, B.H. *Fundamentals of Speech Recognition*; Prentice Hall: Upper Saddle River, NJ, USA, 1993.
11. Zhang, M.; Sawchuk, A.A. Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach. In Proceedings of the ACM SIGHIT International Health Informatics Symposium (IHI), Miami, FL, USA, 28–30 January 2012; pp. 631–640.
12. Fatima, I.; Fahim, M.; Lee, Y.K.; Lee, S. A Unified Framework for Activity Recognition-Based Behavior Analysis and Action Prediction in Smart Homes. *Sensors* **2013**, *13*, 2682–2699.
13. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269.
14. Forney, G.D., Jr. The viterbi algorithm. *Proc. IEEE* **1973**, *61*, 268–278.

15. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning, (ICML'01), Williamstown, MA, USA, 28 June–1 July 2001.
16. Garcia-Ceja, E.; Brena, R. Long-Term Activities Segmentation Using Viterbi Algorithm with a k-Minimum-Consecutive-States Constraint. *Procedia Comput. Sci.* **2014**, *32*, 553–560.
17. Sutton, C.; McCallum, A. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*; MIT Press: Cambridge, MA, USA, 2007; pp. 93–129.
18. Arbelaitz, O.; Gurrutxaga, I.; Muguerza, J.; Pérez, J.M.; Perona, I. An extensive comparative study of cluster validity indices. *Pattern Recognit.* **2013**, *46*, 243–256.
19. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J.M. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors* **2014**, *14*, 10146–10176.
20. Romera-Paredes, B.; Aung, M.S.H.; Bianchi-Berthouze, N. A One-vs-One Classifier Ensemble With Majority Voting for Activity Recognition. In Proceedings of the 21st European Symposium on Artificial Neural Networks, (ESANN 2013), Bruges, Belgium, 24–26 April 2013.
21. Lee, S.W.; Mase, K. Activity and location recognition using wearable sensors. *IEEE Pervasive Comput.* **2002**, *1*, 24–32.
22. Karantonis, D.; Narayanan, M.; Mathie, M.; Lovell, N.; Celler, B. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans. Inf. Technol. Biomed.* **2006**, *10*, 156–167.
23. Mannini, A.; Sabatini, A.M. Machine Learning Methods for Classifying Human Physical Activity from on-Body Accelerometers. *Sensors* **2010**, *10*, 1154–1175.
24. Kwapisz, J.R.; Weiss, G.M.; Moore, S.A. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* **2011**, *12*, 74–82.
25. Banos, O.; Galvez, J.M.; Damas, M.; Pomares, H.; Rojas, I. Window Size Impact in Human Activity Recognition. *Sensors* **2014**, *14*, 6474–6499.
26. Lara, O.; Labrador, M. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1192–1209.
27. Witten, I.; Frank, E.; Hall, M. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed.; The Morgan Kaufmann Series in Data Management Systems; Elsevier Science: Amsterdam, The Netherlands, 2011.
28. Gu, T.; Wu, Z.; Tao, X.; Pung, H.K.; Lu, J. epSICAR: An Emerging Patterns Based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications, Galveston, TX, USA, 9–13 March 2009; pp. 1–9.
29. Cook, D.; Krishnan, N.; Rashidi, P. Activity Discovery and Activity Recognition: A New Partnership. *IEEE Trans. Cybern.* **2013**, *43*, 820–828.
30. Huynh, T.; Blanke, U.; Schiele, B. Scalable Recognition of Daily Activities with Wearable Sensors. In *Location- and Context-Awareness*; Hightower, J., Schiele, B., Strang, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4718, pp. 50–67.

31. Mitchell, E.; Monaghan, D.; O'Connor, N.E. Classification of sporting activities using smartphone accelerometers. *Sensors* **2013**, *13*, 5317–5337.
32. Ugolotti, R.; Sassi, F.; Mordonini, M.; Cagnoni, S. Multi-sensor system for detection and classification of human activities. *J. Ambient Intell. Humaniz. Comput.* **2013**, *4*, 27–41.
33. Lee, Y.S.; Cho, S.B. Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer. In *Hybrid Artificial Intelligent Systems*; Corchado, E., Kurzyński, M., Woźniak, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6678, pp. 460–467.
34. Guenterberg, E.; Ghasemzadeh, H.; Jafari, R. Automatic Segmentation and Recognition in Body Sensor Networks Using a Hidden Markov Model. *ACM Trans. Embed. Comput. Syst.* **2012**, *11*, 46:1–46:19.
35. Van Kasteren, T.L.; Englebienne, G.; Kröse, B.J. An Activity Monitoring System for Elderly Care Using Generative and Discriminative Models. *Pers. Ubiquitous Comput.* **2010**, *14*, 489–498.
36. Vinh, L.; Lee, S.; Le, H.; Ngo, H.; Kim, H.; Han, M.; Lee, Y.K. Semi-Markov conditional random fields for accelerometer-based activity recognition. *Appl. Intell.* **2011**, *35*, 226–241.
37. Huynh, T.; Fritz, M.; Schiele, B. Discovery of activity patterns using topic models. In Proceedings of the 10th International Conference on Ubiquitous Computing, (UbiComp'08), Seoul, Korea, 21–24 September 2008; pp. 10–19.
38. Tolstikov, A.; Hong, X.; Biswas, J.; Nugent, C.; Chen, L.; Parente, G. Comparison of fusion methods based on DST and DBN in human activity recognition. *J. Control Theory Appl.* **2011**, *9*, 18–27.
39. Sung, J.; Ponce, C.; Selman, B.; Saxena, A. Human Activity Detection from RGBD Images. Available online: http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewFile/4000/4315 (accessed on 24 November 2014).
40. Gordon, D.; Hanne, J.H.; Berchtold, M.; Shirehjini, A.; Beigl, M. Towards Collaborative Group Activity Recognition Using Mobile Devices. *Mob. Netw. Appl.* **2013**, *18*, 326–340.
41. Wallach, H.M. *Conditional Random Fields: An Introduction*; Technical Report MS-CIS-04-21; University of Pennsylvania: Philadelphia, PA, USA, 2004.
42. Ratnaparkhi, A. A maximum entropy model for part-of-speech tagging. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 17–18 May 1996; Volume 1, pp. 133–142.
43. GENEActiv Available online: http://www.geneactiv.org/ (accessed on 24 November 2014).
44. Zhang, S.; Murray, P.; Zillmer, R.; Eston, R.G.; Catt, M.; Rowlands, A.V. Activity classification using the GENEA: Optimum sampling frequency and number of axes. *Med. Sci. Sports Exerc.* **2012**, *44*, 2228–2234.
45. Kriete, A.; Eils, R. *Computational Systems Biology: From Molecular Mechanisms to Disease*; Elsevier Science: Amsterdam, The Netherlands, 2013.
46. Cavnar, W.B.; Trenkle, J.M. N-gram based text categorization. In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94), Las Vegas, NV, USA, 11–13 April 1994.

47. Sivogolovko, E.; Novikov, B. Validating Cluster Structures in Data Mining Tasks. In Proceedings of the 2012 Joint EDBT/ICDT Workshops (EDBT-ICDT'12), Berlin, Germany, 26–30 March 2012; pp. 245–250.
48. Xu, R.; Xu, J.; Wunsch, D. A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2012**, *42*, 1243–1256.
49. Pakhira, M.K.; Bandyopadhyay, S.; Maulik, U. Validity index for crisp and fuzzy clusters. *Pattern Recognit.* **2004**, *37*, 487–501.
50. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65.
51. Bezdek, J.; Pal, N. Some new indexes of cluster validity. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1998**, *28*, 301–315.
52. Liu, Y.; Li, Z.; Xiong, H.; Gao, X.; Wu, J. Understanding of internal clustering validation measures. In Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM), Sydney, Australia, 13–17 December 2010; pp. 911–916.
53. Eagle, N.; Pentland, A. Reality mining: Sensing complex social systems. *Pers. Ubiquitous Comput.* **2006**, *10*, 255–268.
54. Shapiro, S.S.; Wilk, M.B. An analysis of variance test for normality (complete samples). *Biometrika* **1965**, *52*, 591–611.
55. Garcia-Ceja, E. Dataset Long-Term Activities. Figshare. Available online: http://dx.doi.org/10.6084/m9.figshare.1029775 (accessed on 24 November 2014).
56. Garcia-Ceja, E. Source Code Long-Term Activities. Figshare. Available online: http://dx.doi.org/10.6084/m9.figshare.1250121 (accessed on 24 November 2014).
57. Himmelmann, L. HMM R Package. Available online: http://cran.r-project.org/web/packages/HMM/index.html (accessed on 1 July 2014).
58. Heilbron, F.C.; Niebles, J.C. Collecting and Annotating Human Activities in Web Videos. In Proceedings of the International Conference on Multimedia Retrieval (ICMR'14), Glasgow, UK, 1–4 April 2014; pp. 377–384.
59. Lasecki, W.S.; Weingard, L.; Ferguson, G.; Bigham, J.P. Finding Dependencies between Actions Using the Crowd. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14), Toronto, ON, Canada, 26 April–1 May 2014; pp. 3095–3098.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).