Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions

Jorge-Luis Reyes-Ortiz[1,2,⋆], Luca Oneto[1], Alessandro Ghio[1], Albert Samá[2], Davide Anguita[1], and Xavier Parra[2]

[1] DITEN – University of Genoa, Via Opera Pia 11A, Genova, 16145, Italy
{Luca.Oneto,Alessandro.Ghio,Davide.Anguita}@unige.it
[2] CETpD - Universitat Politècnica de Catalunya, Vilanova i la Geltrú 08800, Spain
Jorge.Luis.Reyes@estudiant.upc.edu, {Albert.Sama,Xavier.Parra}@upc.edu

Abstract. Postural Transitions (PTs) are transitory movements that describe the change of state from one static posture to another. In several Human Activity Recognition (HAR) systems, these transitions cannot be disregarded due to their noticeable incidence with respect to the duration of other Basic Activities (BAs). In this work, we propose an online smartphone-based HAR system which deals with the occurrence of postural transitions. If treated properly, the system accuracy improves by avoiding fluctuations in the classifier. The method consists of concurrently exploiting Support Vector Machines (SVMs) and temporal filters of activity probability estimations within a limited time window. We present the benefits of this approach through experiments over a HAR dataset which has been updated with PTs and made publicly available. We also show the new approach performs better than a previous baseline system, where PTs were not taken into account.

Keywords: Human Activity Recognition, Smartphones, Postural Transitions, Support Vector Machines, Temporal Filtering.

1 Introduction

Human Activity Recognition is nowadays an active research field which aims to understand human behavior through the interpretation of sensory information gathered from people and the environment they live in: this enables context-awareness, which allows emergence of more interactive and cognitive environments [7,4]. One of the mechanisms to obtain user-related activity information is through wearable sensors. With them, attributes for describing motion, location and physiological signals can be easily and directly collected from the user. However, sometimes they have the disadvantage being obtrusive and restrict

⋆ This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

S. Wermter et al. (Eds.): ICANN 2014, LNCS 8681, pp. 177–185, 2014.

user's movement. For this reason, the use of smartphones for wearable sensing is a interesting alternative that brings significant advantages: *a.)* These devices are already provided with embedded motion sensors (e.g. accelerometer, gyroscope, magnetometer, GPS, etc) that can be used for activity detection, and *b.)* People are nowadays familiarized and more comfortable with smartphones because they continuously interact with these devices throughout the day. These two aspects, combined with their computing characteristics and the possibility of collecting and distributing data, make them a exploitable tool for HAR.

In this work we present a novel smartphone-based online HAR system for the classification of activities, which takes into consideration the impact of PTs in the system performance. Most of the HAR approaches ignore transitions between activities because their occurrence is quite low and duration with respect to other activities is shorter [9]. Nevertheless, this assumption is application-dependent and does not apply in cases where various tasks need to be performed in a short time, such as in the development of monitoring systems for patients during rehabilitation practices. In general, when PTs occur, an online system can manifest fluctuations in the classification as these states are unspecified and therefore reduce its performance. To overcome this problem, we propose a method based on a multiclass Support Vector Machine (SVM) that performs activity probability estimations for each activity. Then, these estimations are interpreted as activity probability signals when combined with the predictions of previous samples, and finally they are heuristically filtered in order to improve classification accuracy during PTs. The proposed method is benchmarked against previous propositions (e.g. [1]). However, though various HAR datasets have been publicly distributed [13,14,5], only few publicly available HAR datasets include smartphone inertial data: therefore, in this paper, the authors contribute by introducing a HAR dataset for research purposes built from the recordings of people performing BAs and also PTs that occur in between them.

2 The HAR Dataset with Postural Transitions

PTs are events with a limited duration. They are characterized by start and end times which usually vary slightly from one person to another. Also, these events are bounded between other two activities and correspond to the transition period between them. Conversely, BAs, such as *standing* and *walking*, can prolong indefinitely. Data collection for these two types of activity is also different: PTs need to be executed repeatedly to get separate samples; instead, BAs are continuous, thus allow many (window) samples to be taken from a single test, limited only by its time extent.

This section contains a detailed description of the HAR dataset. In [1], we presented a publicly available dataset ($\mathcal{D}_0$) for the classification of activities using data gathered from the smartphone inertial sensors, which was made available on the Internet [3]. We have updated this original approach in order to include PTs. The experiment was planned in order to contain six BAs: three static postures (*standing, sitting, lying*) and three ambulation activities (*walking, walking* *downstairs and walking upstairs*). Moreover, it was arranged with the intention of having also available all the possible PTs that occur between the three existing static postures. These are: *stand-to-sit*, *sit-to-stand*, *sit-to-lie*, *lie-to-sit*, *stand-to-lie*, and *lie-to-stand*. We collected signals from the device's embedded triaxial accelerometer and gyroscope at a proper frequency to capture human body motion [7]. Signals were then synchronized with the experiment videos in order to use them as the ground-truth for manual labeling. Finally, the dataset was randomly partitioned into two groups (70% for training and 30% for testing purposes).

The updated dataset[1], from now on referred as $\mathcal{D}_1$, provides some relevant information regarding the duration of activities. By repeating twice each PT, the patients involved in the dataset creation allowed to derive a total of 60 labels for each PT, comprising a non-negligible 9% of the entire recorded time of the experimental data. In particular, PTs have an average duration of 3.73s $\pm$ 1.17 seconds, though this duration varies on each type of PT. We also found that inverse transitions such as *lie-to-sit* and *sit-to-lie* have different average durations. Moreover, some PTs can be described as the sequence of other two (e.g. *stand-to-lie* combines *stand-to-sit* and *sit-to-lie* PTs), as it can be observed on the experiment videos. In the particular case of *sit-to-stand* and *stand-to-sit* PTs, our finding matches the measurements of the average duration of these transitions performed by healthy patients in [10].

On the other hand, we found that the execution of the rest of activities (BAs) took longer times (17.3s $\pm$ 5.7 in average). This finding is important given that, in general, the execution time of activities in real life takes longer than PTs which have nearly-fixed duration. As a consequence, this helps us to define the heuristic rules to filter PTs in the proposed recognition system.

3 The Method

The entire recognition algorithm is composed of three modules. The first one (MOD1) comprises data acquisition and signal conditioning from the inertial sensors to obtain the features that characterize each activity sample. These features become the input of the prediction module (MOD2), *SVM with probability estimates*, where they are evaluated and, from each sample, the probabilities of belonging to each of the 6 BAs are estimated. These probabilities are then joined in the last module (MOD3) along with the predictions of previous activity window samples and processed by means of temporal filtering. This is achieved by applying a set of defined heuristic filters aiming to avoid fluctuations in the classification of BAs. These filters also introduce the *unknown activity* (UA) class when BA probabilities are marginal. Lastly, probabilities are used to define the most likely activity at a given time t. This is combined with further pruning of the discrete output through filtering using a rule-based approach. Algorithm 1 depicts the whole recognition process and its main modules are described as follows.

MOD1 takes as input the triaxial linear acceleration $\boldsymbol{a}_{raw}(t)$ and angular velocity $\boldsymbol{\omega}_{raw}(t)$ time signals. Signal conditioning consists of first applying noise

[1] The dataset is currently available at `http://har.smartlab.ws`.

Algorithm 1. HAR Method

Require: $\boldsymbol{a}$: Triaxial linear acceleration, $\boldsymbol{\omega}$: Triaxial angular velocity, $\boldsymbol{g}$: Gravity, $H_1(\cdot)$: Noise reduction transfer function, $H_2(\cdot)$: Body acceleration transfer function, $\phi(\cdot)$: Feature extraction function, T: Windows size, d: Number of classes, n: Buffer length, B: Buffer of probability vectors $B \in \mathbb{R}^{d \times n}$, B': Filtered buffer of probability vectors, $\boldsymbol{z}$: Buffer of discrete activity predictions $\boldsymbol{z} \in \mathbb{R}^n$, $\Phi(\cdot)$: Probability filtering function, $\Psi(\cdot)$: Discrete filtering function

function $(\boldsymbol{a}(t), \boldsymbol{g}(t), \boldsymbol{\omega}(t))$ = ProcessInertialSignals($\boldsymbol{a}_{raw}(t), \boldsymbol{\omega}_{raw}(t)$)

- $\boldsymbol{a}_{total}(t) = H_1(\boldsymbol{a}_{raw}(t))$, // Noise Filtering
- $\boldsymbol{\omega}(t) = H_1(\boldsymbol{\omega}_{raw}(t))$, $\boldsymbol{a}(t) = H_2(\boldsymbol{a}_{total}(t))$ // Body acceleration Extraction
- $\boldsymbol{g}(t) = \boldsymbol{a}_{total}(t) - \boldsymbol{a}(t)$ // Gravity extraction

end

function α = OnlinePrediction($t, \boldsymbol{a}(t), \boldsymbol{g}(t), \boldsymbol{\omega}(t), B, \boldsymbol{z}$)

- $A = \{\boldsymbol{a}(t')\}$, $G = \{\boldsymbol{g}(t')\}$, $\Omega = \{\boldsymbol{\omega}(t')\}$, $t' \in [t - T, \ldots, t]$ // Window sampling
- $\boldsymbol{x} = \phi(A, G, \Omega)$ // Feature Extraction and Normalization
- $\boldsymbol{p} = []$
- **for** $c \in \{1, \ldots, d\}$ **do** // Multiclass SVM
  - $\boldsymbol{p} = \left[\boldsymbol{p} \mid 1/\left(1 + e^{\left[\Gamma^c\left(\boldsymbol{w_c}^T \boldsymbol{x} + b_c\right) + \Delta^c\right]}\right)\right]$ // FFP with probability estimation
- **end**
- $B = \{\boldsymbol{p}^T \mid B(1 : end - 1, :)\}$ // Append probability vector
- $B' = \Phi(B)$ // Activity probability filtering
- $\hat{\theta}_{MAP} = \arg\max_{c \in [1, \ldots, d]} \quad b'_{(n-1,c)}$ // MAP
- $\boldsymbol{z} = \{\hat{\theta}_{MAP} \mid \boldsymbol{z}(1 : end - 1)\}$ // Append last activity prediction
- $\alpha = \Psi(\boldsymbol{z})$ // Discrete filtering and activity estimation

end

reduction filters (with transfer function is represented by $H_1()$). After that, clean triaxial acceleration $\boldsymbol{a}_{total}(t)$ and angular velocity $\boldsymbol{\omega}(t)$ signals are obtained. The acceleration signal is further processed and separated into body motion acceleration $\boldsymbol{a}(t)$ and gravity $\boldsymbol{g}(t)$. A detailed description of this module is described in [1].

The signal conditioning process is continuously executed over the inertial signals as represented in the $ProcessInertialSignals()$ function in Algorithm 1. In addition, the $OnlinePrediction()$ function is in charge of the recognition of activities and it is periodically executed to obtain and classify window samples (A, G, Ω) from the filtered inertial signals. Its periodicity satisfies the sliding-windows criteria: a time span of 2.56s and 50% overlap between them. Features are extracted from these window samples through measures in the time and frequency domain (represented by the function $\phi()$). They provide a collection of 561 informative features which has been selected based on previous works in the literature [1].

MOD2 consists of a set of 6 One-Vs-All (OVA) binary SVMs [6,12] which characterize each of the one studied activities. This algorithm is comprehensively described in [2]. In particular, if we consider a dataset composed of l patterns of pairs $(\boldsymbol{x}_i, y_i)$ $i \in \{1, ..., l\}$, $\boldsymbol{x}_i \in \mathbb{R}^m$, and $y_i = \{\pm 1\}$, a binary SVM model can be identified by solving a Convex Constrained Quadratic Programming (CCQP) minimization problem:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \qquad \text{s.t.} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \boldsymbol{C} \quad \boldsymbol{y}^T \boldsymbol{\alpha} = 0 \tag{1}$$

where the C hyperparameter is the regularization term and the matrix $Q \in \mathbb{R}^{l \times l}$ is defined such that $q_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$. The prediction of new patterns can be achieved with the SVM Feed Forward Phase (FFP) given by $f(\boldsymbol{x}) = \sum_{i=1}^{l} y_i \alpha_i \boldsymbol{x}_i^T \boldsymbol{x}_j + b =$

$\boldsymbol{w}^T\boldsymbol{x} + b$, where the bias term b is obtained with the method proposed in [8]. As, in this case, we are dealing with a multiclass problem, for each binary classifier we have to compute probability estimates $p_c(\boldsymbol{x})$ which represent how likely is for a new sample pattern to be classified as a given class. For a given number of classes d and a test sample $\boldsymbol{x}$, the probability output of each SVM ($p_c(\boldsymbol{x})\ \forall\ c \in [1, ..., d]$) is compared against the others to find the class c^* with the Maximum A–Posteriori Probability (MAP). Assuming that all the classes have the same a priori distribution then: $c^* = \arg\max_c p_c(\boldsymbol{x})$. The probability estimation method we employ was proposed by Platt in [11] and uses the predicted FFP output of the training set and its ground-truth label to fit a sigmoid function of the form: $p(\boldsymbol{x}) = \left(1 + e^{(\Gamma f(\boldsymbol{x}) + \Delta)}\right)^{-1}$, where Γ and Δ are the function parameters which optimal values can be found using the error minimization function:

$$\arg\min_{\Gamma,\Delta} \sum_{i=1}^{l} y_i \log p(\boldsymbol{x}_i) + (1 - y_i)\log(1 - p(\boldsymbol{x}_i)) \tag{2}$$

The last module (MOD3) is composed of a series of filtering processes. The standard SVM classifier produces only a discrete output that indicates the class that best represents a test input (window sample). Moreover, this classifier is a static method which only depends on its input $\boldsymbol{x}$ and is not affected, for instance, by other factors such as previous samples or how probable the other activities are while running the FFP. Considering that in real world situations activities can be described as a sequence of correlated events, we take advantage of the SVM with probability estimates in a more extensive way. Indeed, the SVM probability predictions for all the activities $\boldsymbol{p} = \{p_1, \ldots, p_d\}$ can be interpreted as an activity probability signal in time when combined with the predicted output from previous samples. This assumption can improve the recognition system as we can exploit signal processing techniques, such as filtering, to make the overall classification performance more robust. We take into account aspects such as the interrelationship within activities and the fact that only one activity happens at a time (e.g. including transitions, which is one of our areas of interest). We have developed a set of heuristic filters to improve the probabilistic output of the SVM using temporal information from each prediction and its neighboring samples. This process is divided in two parts: *Probability Filtering*, which directly handles the probability signals, and *Discrete filtering* that further filters the activity output after the discretization of probabilities into activities.

For the first part, the implemented filters are a rule-based approach that uses as input the matrix of probability vectors B whose columns are the SVM predictions of the last n overlapped windows. The largest filter requires $n_{max} = 5$, which is equivalent to a prediction delay of 5.12s. They use probability thresholds to define whether a class is considered active (e.g. $p_c > threshold$) or to condition the filtering of an activity based on the value of other classes. $B' = \Phi(B)$ represents the application of the probability filters over the activity sequence. Two types of probability filters were used: the *Transition Filter* is aimed to remove peaks and transients of dynamic activities when they appear amongst static ones. This is applied to PTs as they exhibit a spiky behavior and usually take a short time (from

2 to 3 seconds): therefore this filter measures the length of the activation of these dynamic signals for a number of overlapping windows (maximum 3). The *Smoothing Filter* targets, on the other hand, the probability signals during the occurrence of BAs. It helps to stabilize signal fluctuations when their probability values are greater than a threshold (0.2) within the activity sequence. This is aimed to make evident, in a sequence of window samples, small differences between activities with high interclass misclassification e.g. *standing* and *sitting* or *walking* and *walking upstairs*. Oscillations are smoothed using a linear interpolation.

The second part (Discrete Filtering) defines the most likely activity for each window sample $\hat{\theta}_{MAP}$. This is done using MAP over the probability vector $b'_{(n-1,:)}$ of the filtered activity sequence matrix B'. It selects one of the classes as the predicted activity. However, sometimes it happens that all the probability values are low. This indicates that none of the classes seems to be representative of the current activity. For this, we have defined a minimum activity threshold which is used to label samples as undefined (UA) when none of the classes reaches this value. This is particularly useful when PTs occur as they are not learned by the SVM model. But in general, this approach can be beneficial in real life situations when the HAR system is used while activities outside the studied set occur. These will not be categorized as any of learned activities, instead the system will show that an unknown event has occurred. This filter removes sporadic activities that appear for a short time and are unlikely to happen for only a single window sample. It also includes cases when the UA is detected and its contiguous activities belong to the same class. The filter allows to relabel them as their neighbors. As a consequence, the final predicted activity is the result of this discrete filter $\alpha = \Psi(\boldsymbol{z})$, where $\boldsymbol{z}$ in the buffer containing the last 3 predicted activities $\hat{\theta}_i$.

4 Experimental Results and Discussions

To evaluate the error of the approach at runtime, some new considerations are required given that PTs are taken into account. The system is expected to avoid fluctuations or activity misclassifications during the occurrence of PTs, either by detecting a UA or by preserving the class of the activities adjacent to each PT. Therefore, in Table 1 we propose a new error assessment method for PTs and BAs. Notice that it considers all the possible activity combinations. Moreover, the selected metric also penalizes unwanted conditions in BAs such as the appearance of the UA class in the classification as these activities are learned in the SVM and should be correctly classified. To evaluate the performance of the new algorithm, we took the original offline HAR system presented in [1] as a reference point which is similar to the proposed method without temporal

Table 1. Classification error assessment. A = Activity, U = Unknown.

Basic Activities			**Postural Transitions**		
Ground-Truth	**Prediction**	**Evaluation**	**Ground-Truth**	**Prediction**	**Evaluation**
A1 - A1 - A1	A1 - A1 - A1	Correct	A1 - PT - A2	A1 - A1∨A2 - A2	Correct
A1 - A1 - A1	A1 - A2 - A1	Incorrect	A1 - PT - A2	A1 - A3 - A2	Incorrect
A1 - A1 - A1	A1 - UA - A1	Incorrect	A1 - PT - A2	A1 - UA - A2	Correct

activity filtering ($\Phi(\cdot)$ and $\Psi(\cdot)$). First of all, we learned the original dataset ($\mathcal{D}_0$) which only considered the 6 BAs and achieved a system error of 3.59%. Then, we applied the same procedure with the updated dataset ($\mathcal{D}_1$), the one with the labeled postural transitions, obtaining an error of 7.72%. This showed an increase of the system error by 4.13% percentage points mainly due to the misclassifications that occurred in PTs. This finding show how the first offline approach fails to work online when it is under a large number of transitory events such as PTs, that in the dataset cover nearly 9% of the data. Although this is a rather small portion, it is influential in the overall system performance.

Henceforth, we separately considered the effect of PTs and BAs in the system and work only with $\mathcal{D}_1$. Table 2 presents the the accumulated error of the HAR approach including intermediate stages of the processing. In this way, it is possible to have an idea how the different stages of the algorithm are progressively affecting the overall classification accuracy. The three stages are: No filtering or just the SVM output (–), Probability Filtering (Φ), and Discrete filtering (Φ, Ψ). Every row represents the stages of the algorithm and the columns the type of activity (BAs, PTs and Combined). From the table, it can be also noticed that the error without filtering is the highest achieved (7.72%). As we decompose this, we can also see that this is mainly due to a large error of 41.34% when classifying PTs. BAs instead remain much lower with a 4.46%. We can also observe that the temporal activity filters widely improve the classification of PTs reaching a minimum error of 5.77%. BAs instead improve only slightly after filtering. The final error achieved was 3.34% which is even lower that what we had obtained with offline approach and the dataset that did not take into account PTs.

In Table 2 we can find the confusion matrices of the system classification before ($\mathcal{C}_1$) and after ($\mathcal{C}_2$) activity temporal filters. For clarity, in the column of predictions of PTs we find true positives even if PTs are not predicted by the SVM: this is because we have relabeled as PT the samples correctly predicted based on the error metric we defined in Table 1 (e.g. predicted as UA or as an adjacent BA). In this way we make the confusion matrix still informative and preserve in its diagonal the correct classifications. Moreover, notice that $\mathcal{C}_2$ is not squared. The number of classes of the ground truth (7 rows: 6 BAs + 1 PT class which combines the 6 available PTs) differs from the number of predicted outputs where the UA was added. In this last column, we allocate the correct predictions of UA during PTs so they do not appear as misclassifications outside the diagonal. In these matrices it is also evident the effects of filtering in the system. For example, the number of false negatives for the PT class in $\mathcal{C}_1$ is quite large but reduced after temporal

Table 2. System error based on filtering stage, type of activity and confusion matrices before and after filtering

Filters	**BAs**	**PTs**	**Overall**
–	4.46%	41.34%	7.72%
Φ	3.45%	18.24%	4.76%
Φ, Ψ	3.10%	5.77%	3.34%

Filters	$\mathcal{C}_1$: –							$\mathcal{C}_2 : \Phi(\cdot),\ \Psi(\cdot)$							
Activity	WK	WU	WD	SI	ST	LD	PT	WK	WU	WD	SI	ST	LD	PT	UA
WK	**542**	0	3	1	0	0	0	**545**	0	0	0	0	0	0	1
WU	32	**523**	2	0	2	0	0	28	**514**	1	0	1	0	0	15
WD	3	4	**498**	0	4	0	0	0	0	**505**	0	3	0	0	1
SI	0	4	0	**481**	71	1	0	0	0	0	**504**	52	0	0	1
ST	3	3	0	18	**588**	0	0	0	1	0	1	**610**	0	0	0
LD	0	0	0	0	0	**604**	0	0	0	0	0	0	**604**	0	0
PT	12	101	1	18	4	0	**193**	0	10	0	9	0	0	**310**	0

filtering, in particular for the dynamic activities (e.g. in *walking upstairs*), which provide most of these misclassifications. It is also worth noting the reduction of interclass misclassifications between similar activities such as in the static postures *sitting* and *standing*, and also between *walking* and *walking upstairs*. Even the false negatives of the *standing* class produced by *sitting* misclassifications are nearly zero; however, the opposite case still preserves some errors.

References

1. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: A public domain dataset for human activity recognition using smartphones. In: European Symposium on Artificial Neural Networks (2013)
2. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. Journal of Universal Computer Science 19, 1295–1314 (2013)
3. Bache, K., Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
4. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
5. Dernbach, S., Das, B., Krishnan, N., Thomas, B., Cook, D.: Simple and complex activity recognition through smart phones. In: International Conference on Intelligent Environments (2012)
6. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. EEE Transactions on Neural Networks 13, 415–425 (2002)
7. Karantonis, D.M., Narayanan, M.R., Mathie, M., Lovell, N.H., Celler, B.G.: Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine 10, 156–167 (2006)
8. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to platt's smo algorithm for svm classifier design. Neural Computation 13, 637–649 (2001)
9. Lara, O., Labrador, M.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials 1, 1–18 (2012)
10. Najafi, B., Aminian, K., Loew, F., Blanc, Y., Robert, P.A.: Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly. IEEE Transactions on Biomedical Engineering 49, 843–851 (2002)
11. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers (1999)
12. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141 (2004)
13. Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A.: Collecting complex activity data sets in highly rich networked sensor environments. In: International Conference on Networked Sensing Systems 2010 (2010)
14. Tapia, E.M., Intille, S.S., Lopez, L., Larson, K.: The design of a portable kit of wireless sensors for naturalistic data collection. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, pp. 117–134. Springer, Heidelberg (2006)