Workshop Proceedings of the 8th International Conference on Intelligent Environments
J.A. Botía et al. (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-080-2-94

Real Time Activity Recognition Using a Cell Phone's Accelerometer and Wi-Fi

Enrique A. GARCIA[1] and Ramón BRENA[2]
ITESM Campus Monterrey

Abstract. In this paper, we present an implementation of a real time activity recognizer running on a cellphone. First, simple activity recognition from accelerometer data is performed and then, this information is fused with data from Wi-Fi Access Points to classify the activity being performed by the user. The training set consisted of 8 activities performed in an academic environment and the classification accuracy was 89.7% using a supervised learning approach.

Keywords. activity recognition, accelerometer, Wi-Fi, cell phone.

Introduction

Human activity recognition is an important task for ambient intelligence systems [1]. 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 security, abnormal behavior can be detected and thus used to prevent a burglary or other criminal activities.

In recent years simple human activity recognition has been achieved successfully, however complex activity recognition is still challenging and is an active area of research. In [2] they pose the following challenges regarding the nature of human activities: *Recognizing concurrent activities, recognizing interleaved activities, ambiguity of interpretation* and *multiple residents*.

In this work we present an implementation of a real time activity recognizer running on a cellphone. First, the *simple activity* is recognized from the cellphone's tri-axial accelerometer and then this information is fused with data gathered from Wi-Fi Access Points to allow a better understanding of the user's context.

The number of Wi-Fi Access Points around the world has increased significantly in the last years. They are installed in many places such as restaurants, hotels, schools, parks, airports, etc. Since every Access Point has a unique identifier namely, the BSSID (Basic Service Set Identifier), it is possible to use this information along with the signal strength for localization and tracking purposes [3,4,5].

The objective of this work is not to infer the spatial location of the user but to determine the user's context by means of a supervised learning approach. For example, if *walking* is detected as a simple activity and the Wi-Fi yields information that tells us that the user is in a library, then we can classify the whole activity as *looking for a book*

[1] E-mail: A00927248@itesm.mx
[2] E-mail: ramon.brena@itesm.mx

and another application can use this information maybe to suggest related titles. If the simple activity is *resting* the whole activity may be classified as *reading a book*. In this case the user's cell phone could block unimportant incoming calls to avoid interruptions. The role of the Access Points is to aid in the discrimination process providing approximate location information, i.e., we can use a fixed set of *primitive activities* as the basis and combine them with 'location' information to generate *contextualized activities.*

This paper is organized as follows. Section 1 presents several recent works in simple and complex activity recognition. Section 2 describes the methodology which includes: data collection, training, and classification. In section 3 we describe the details of the experiment and its results. Section 4 presents different approaches for human activity recognition. Finally, in section 5 conclusions and future work are presented.

1. Related Work

In this section we present a survey of works that tackle the problem of recognizing simple activities (*walking*, *running*, *sitting down*, *falling*, etc.) and complex activities (*making coffee*, *cleaning the house*, *making a drink*, *having dinner*, etc.).

1.1. Simple Activities Recognition

Generally, simple activities do not depend on the context, i.e., they can exist by themselves and they last only a few seconds. Examples of this type of activities are: *running*, *walking*, *resting*, *sitting*, etc.

Brezmes, Gorricho and Cotrina [6] implemented a real time activity recognizer on a mobile phone. They achieved accuracies ranging from 70% to 90% for several activities. Mannini and Sabatini [7] used five bi-axial accelerometers located at the hip, wrist, arm, ankle, and thigh and they reported accuracies between 93% and 98.5% for seven different activities (*sitting*, *lying*, *standing*, *walking, stair climbing*, *running* and *cycling*). Karantonis, *et al*. [8] presented an implementation of a real time activity classifier capable of computing the metabolic energy expenditure. Ravi, *et al*. [9] made a comparison of base-level classifiers and meta-level classifiers and concluded that combining classifiers using Plurality Voting turned out to be the best choice for the recognition of simple activities. Mi Zhang [10] proposed a Bag-of-Features approach which builds activity models using histograms of primitive symbols. Recently, the Bag-of-Features approach has gained significant interest.

1.2. Complex Activities Recognition

Complex activities are composed of a collection of simple activities and may consider information from the context, time, and interactions between other persons and objects. The recognition of these activities generally requires more sensors and a fixed infrastructure (video cameras, RFID tags, several accelerometers, magnetic sensors, etc.).

Tao Gu, *et al.* [11] built activity models by mining a set of Emerging Patterns from a sequential activity trace and used them to recognize sequential, interleaved, and concurrent activities achieving accuracies of 90.96%, 87.98% and 78.58%, respectively. Tam Huynh, *et al.* [12] used topic models to recognize activities such as: *dinner*, *commuting*, *lunch* and *office work*. They automatically extract activity patterns from sensor data (3D accelerometer, clock, binary tilt switches, temperature sensor, and two light sensors) to enable the recognition of daily routines as a composition of such activity patterns. Experimental results obtained by Tam Huynh, *et al.* [13] suggest that the recognition of complex activities can be achieved with the same algorithms of simple activities. The complex activities they recognized were *preparing for work*, *going shopping* and *doing housework*. Tian, *et al.* [14] use accelerometer and GPS information to automatically send updates to a micro-blogging website. They used Hidden Markov Models for the activity recognition and increased the accuracy by constraining the context using GPS location data.

The approach we use lies between simple and complex activities in the sense that we first recognize the simple activity and then we add Wi-Fi information to contextualize it. This work differs from the previously mentioned in the following aspects. First, we take advantage of existing infrastructure (Wi-Fi Access Points) so our approach does not require the addition of sensors to the environment like RFID tags, video cameras, etc. Second, aside from the presence of in range Access Points, we do not need a fixed configuration of the environment. Since we are just reading the BSSID and signal strength to classify the activities, we do not need to configure each Access Point neither know their location. Finally, we focused in using sensors that are commonly available in most smartphones so the user is freed from having to wear several sensors attached to his/her body.

2. Method

In this section we describe the process of data collection, training, and the activity classification phase. From now on, by *primitive* or *simple activity* we mean the type of activities described in section 1.1 such as *running*, *walking*, *resting*, *sitting,* etc. and by *contextualized activity* we mean an activity that is composed by a *simple activity* plus the information gathered from Wi-Fi Access Points.

2.1. Data Collection

An Android 2.2 [15] application running on a LG Optimus Me cell phone was used to collect the data from the accelerometer and Wi-Fi.

The cellphone was placed in the user's belt and the data collection consisted of 8 activities: *1)reading in bedroom A, 2)watching television, 3)reading in bedroom B, 4)sitting in the lobby, 5)reading in the library (first floor), 6)looking for a book in the library (first floor), 7)reading in the library (second floor), 8)looking for a book in the library (second floor).* Activities 1-4 were performed in an apartment building while activities 5-8 were performed in a library. The data collection process was performed by two participants under supervision. The test set was collected independently in a different day from the training set.

From the tri-axial accelerometer sensor, we read the acceleration values from each of the axes (*x,y,z*) and classify the *primitive activity* being performed as one of *walking, running* or *resting.* There are several approaches for recognizing primitive activities (see section 1.1). For this work we used a nearest neighbor approach [6].

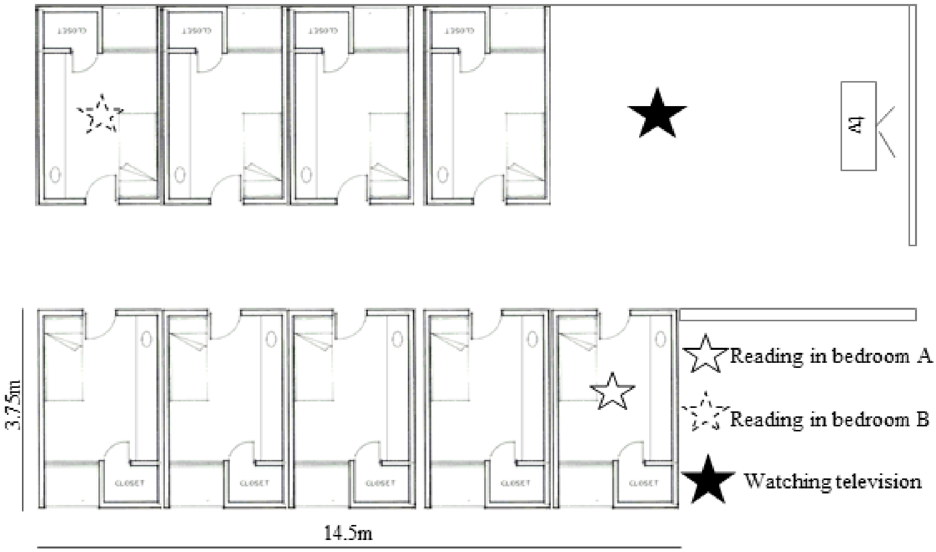

Figure 1. Layout of the apartments building 3rd floor.

From the Wi-Fi sensor, we collected data from the in range Wireless Access Points. Specifically, we collected their BSSID (Basic Service Set Identifier) and signal strength. We selected activities that are very close to each other. Figure 1 shows the layout of the 3rd floor of the apartments building. The lobby is located below the room marked with *Reading in bedroom A* but in the 1st floor.

2.2. Training

In the training phase we generate the instances that will be used as the training set. An instance based learning algorithm (K-nearest neighbors) [16] was used for the *contextualized activities* recognition. Each activity instance has 3 attributes (Table 1).

Figure 2 shows the process of generating one training instance. First, the application performs two scans to collect the data of the in range Access Points. A delay of 500*ms* is set between the scans. Then, every second the *primitive activity* being performed by the user is recognized and stored in a vector *V*. This is done during 5 seconds, i.e., at the end of the 5 seconds, the vector *V* will contain 5 ids' (one for each detected *primitive activity*). Finally the application performs two more scans to gather information from the Access Points. The reason of doing several scans is because in [17] they observed that sometimes one or more Access Points may not be detected because limited sensitivity of the hardware and/or long beacon interval of some Access Points.

Now the instance is created and its *primitive activity* is set to Mode(*V*), i.e., the *primitive activity* that was dominant across the 5 seconds period. For every Access Point found during the scans a pair <*bssid,strength*> is added to *L*, where *bssid* is the Access Point identifier and *strength* is the mean of the signal strength from the 4 scans.

Table 1. Instance Attributes.

Name	Description
Id	Unique identifier of the instance (for debugging purposes)
Class	A number from 1 to 8 to identify which *contextualized activity* the instance belongs to.
Primitive activity	Identifies the *simple activity* associated with this *contextualized activity*. 0 for walking, 1 for running and 2 for resting
List of Access Points	A list L in which each element is a pair *<bssid,strength>*

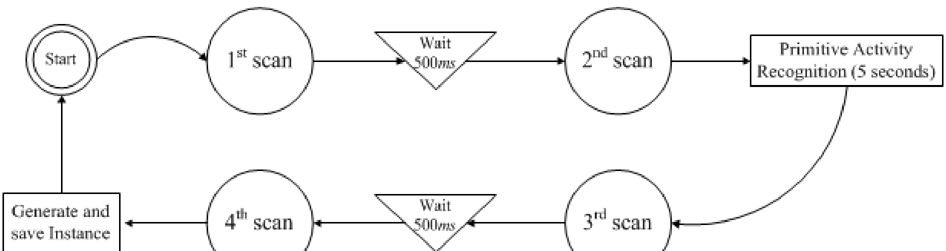

Figure 2. Steps to generate one training instance

2.3. Classification

The K-Nearest Neighbors (K-NN) method was used for the classification. The method consists of computing the distance between the query instance and every other instance from the training set and selecting the *k* nearest instances. For this work we used the Euclidean distance:

$$\sqrt{\sum_{i=1}^{n}\left(b_i - c_i\right)^2} \tag{1}$$

where n is the number of attributes, b is the value of the i^{th} attribute of the query instance and c is the value of the i^{th} attribute of the training instance. For the experiments we used $k = 3$. The differences $(b_i - c_i)$ for $i=1..3$ were computed as follows:

- Primitive activity id. Set to 0 if both b and c have the same primitive activity.

$$dif(b_1, c_1) = \begin{cases} 0 \ if \ P(b) = P(c) \\ 1 \ otherwise \end{cases} \quad (2)$$

where the function P returns the primitive activity associated with the specified instance.

- Ratio of same Access Points. The extreme cases are when both instances share the same Access Points (in this case the distance is 0) and when they do not have any common Access Point (in this case the distance is 1).

$$dif(b_2, c_2) = 1 - (same / total) \quad (3)$$

where $total = cardinality\{L(b) \cup L(c)\}$ and the function L returns the list of Access Points of the specified instance. Similarly, the variable *same* is computed as $same = cardinality\{L(b) \cap L(c)\}$. Eq. (3) is known as the Jaccard distance.

- Difference of the signal strength's standard deviation. This is defined as:

$$dif(b_3, c_3) = 1 - (1/1 + \alpha) \quad (4)$$

where $\alpha = abs(SD(a,b) - SD(b,a))$ and $SD(p1,p2)$ is a function that returns the standard deviation of the signal strength of all Access Points of *p1* that are also in *p2*.

For the real time classification, a query instance is crated in the same way a training instance is created (see Figure 2) and then it is classified using K-NN algorithm.

3. Experiments

In this section we describe the details of the experiments and then we present the results of the classifications.

3.1. Experiment Description

For this experiment we collected a total of 741 instances for the training set and 243 instances for the test set. Tables 2 and 3 show the number of instances per activity and the average number of detected Access Points. The average number of detected Access Points is important because in [17] they observed that the number of received Access Points strongly affects accuracy of proximity classification. They improved their results by performing three scans and feeding the algorithm with the scan that has the highest number of detected Access Points.

Table 2. Number of training instances and their respective average number of detected Access Points

Composed Activity	Number of training instances	Average number of detected Access Points
1) Reading in bedroom A	102	2.2
2) Watching television	104	4.0
3) Reading in bedroom B	68	1.5
4) Sitting in the lobby	91	5.1
5) Reading in the library (first floor)	107	4.2
6) Looking for a book in library (first floor)	78	8.5
7) Reading in the library (second floor)	103	4.9
8) Looking for a book in library (second floor)	88	10.8

Table 3. Number of test instances and their respective average number of detected Access Points

Composed Activity	Number of test instances	Average number of detected Access Points
1) Reading in bedroom A	29	2.3
2) Watching television	28	3.7
3) Reading in bedroom B	27	1.5
4) Sitting in the lobby	32	4.2
5) Reading in the library (first floor)	36	6.0
6) Looking for a book in library (first floor)	30	7.0
7) Reading in the library (second floor)	31	9.9
8) Looking for a book in library (second floor)	30	8.4

The time taken to complete the process shown in Figure 2 for each generated instance takes about 7 seconds, i.e., 5 seconds for the *primitive activity* recognition and approximately 2 seconds for the 4 Wi-Fi scans. Each scan takes less than 1 second but it varies depending on the number of visible Access Points, hardware and other physical issues.

Table 4. Confusion Matrix

Activity	Classified as							
	Reading in bedroom A	Watching television	Reading in bedroom B	Sitting in the lobby	Reading in library 1f	Looking for book in library 1f	Reading in library 2f	Looking for book in library 2f
Reading in bedroom A	**22**	5	0	2	0	0	0	0
Watching television	0	**27**	1	0	0	0	0	0
Reading in bedroom B	5	3	**19**	0	0	0	0	0
Sitting in the lobby	0	0	0	**32**	0	0	0	0
Reading in library 1f	0	0	0	0	**32**	0	4	0
Looking for book in 1f	0	0	0	0	0	**26**	0	4
Reading in library 2f	0	0	0	0	0	0	**31**	0
Looking for book in 2f	0	0	0	0	0	1	0	**29**

3.2. Results and Discussion

The overall classification accuracy using holdout validation was 89.7%. Table 4 shows the confusion matrix. From this table we can see, e.g., that 1 of the 28 instances of *watching television* activity was misclassified as *reading in bedroom B* and the remaining 27 instances were correctly classified. The main diagonal shows the number of correctly classified instances. We performed 10-fold cross validation over the entire data set (984 instances = training + test instances) and the resulting accuracy was 90.3%.

It can be seen that there is a relation between the average number of detected Access Points per activity (Table 1, 2) and the accuracy of the classification. For example, *reading in bedroom B* has the lowest average of detected Access Points (just 1), and in the confusion matrix it can be seen that 8 of its instances were misclassified. In contrast, the activities with high number of detected Access Points had fewer misclassifications. Then, based on these results it appears that in order to achieve good classification accuracies for this experiment's configuration, the activity must have at least an average of 3 detected Access Points.

The major source of error was between activities that are physically close to each other. Given that the library is very far from the departments building there were no misclassifications between these activities.

4. Approaches for Human Activity Recognition

Table 5 shows that due to its simplicity and accuracy accelerometers are the main sensors used in activity recognition. For *complex activities* more sensors (RFID, clock, temperature sensor, light sensors, etc.) are required. As mentioned in section 1 this work uses an approach that lies between simple and complex activities because we first recognize a simple activity and then we add Wi-Fi information to contextualize it.

Table 5. Different approaches for human activity recognition

Reference	Activity Type	Sensors	Approach	No. Activities	Accuracy [%]
[6]	Simple	1 tri-axial accelerometer	KNN	6	70-90
[7]	Simple	5 bi-axial accelerometers	HMM	7	93-98.5
[8]	Simple	1 tri-axial accelerometer	Decision Tree	12	90.8
[10]	Simple	1 tri-axial accelerometer, 1 tri-axial gyroscope	Bag of Features	9	92.7
[18]	Simple	Video sequences	Bayesian Classifier	5	1.5 error rate
Our approach	Contextualized	1 tri-axial accelerometer, Wi-Fi	KNN	8	89.7
[11]	Complex	3 iMote2 sets, 2 RFID wristband readers	Emerging Patterns	26	78.58-90.96

[12]	Complex	3D accelerometer, real time clock, 9 binary tilt switches, temperature sensor, 2 light sensors	Topic Models	34	72.7
[13]	Complex	2D accelerometer, 9 binary tilt switches	K-means, SVM, Nearest Neighbor, HMM	3	80.6-91.8

5. Conclusions and Future Work

In this work we presented a way to combine data from an accelerometer and Wi-Fi to recognize different types of activities. Using simple supervised learning algorithms we achieved good results in the classifications accuracy, subject to the number of detected Access Points and the proximity between the activities.

We chose activities that are common in academic environments, however they may not generalize well for a wider range of activities involving other configurations, e.g., office activities, sport activities, home activities, etc.

For this work the *primitive activities* were chosen by getting the mode activity from the 5 second period. Just taking the mode imposes a limit on the number of activities that can be detected within a given area since they may overlap.

The next step of this research is to devise a way of generating a *contextualized activity* that can include more than 1 *simple activity*, i.e. instead of setting the primitive activity attribute just as the mode we want to use the complete sequence of *simple activities* and take the sequence order into account. For the next phase of this research, we will include a wider range of activities in order to tell whether this method is suitable for various situations and configurations.

References

[1] D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. In: Pervasive and Mobile Computing, vol. 5(4) pp. 277–298 (2009)

[2] E. Kim, S. Helal, and D. Cook. Human Activity Recognition and Pattern Discovery. IEEE Pervasive Computing, vol. 9(1) pp. 48–53 (2010)

[3] J. Krumm, and E. Horvitz. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In: Mobiquitous, pp. 4–13. (2004)

[4] J. Correa, E. Katz, P. Collins, and M. Griss. Room-Level WiFi LocationnTracking. Carnegie Mellon Silicon Valley, CyLab Mobility Research Center technical report MRC-TR-2008-02 (2008)

[5] G. Zaruba, M. Huber, F. Kamangar, and I. Chlamtac. Monte Carlo Sampling Based In-home Location Tracking With Minimal RF Infrastructure Requirements. In: Proceedings of IEEE GLOBECOM, pp. 3624–3629 (2004)

[6] T. Brezmes, J. L. Gorricho, and J. Cotrina. Activity Recognition from Accelerometer Data on a Mobile Phone. In: Proceedings of the IWANN '09, pp. 796–799. Springer, (2009)

[7] A. Mannini, and A. M. Sabatini. Machine learning methods for classifying human physical activity from on body accelerometers. In: Sensors, vol. 10(2) pp. 1154–1175 (2010)

[8] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine, vol. 10 pp. 156–167 (2006)
[9] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity Recognition from Accelerometer Data. In: AAAI, pp. 1541–1546 (2005)
[10] M. Zhang, and A. Sawchuk. Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium (IHI '12). ACM, New York, NY, USA, pp. 631-640 (2012)
[11] T. Gu, Z. Wu, X. Tao, H. Pung, and J. Lu. epSICAR: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In: 7th Conf. on Pervasive Computing and Communications. IEEE P., New York (2009)
[12] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In: Proceedings of the International Conference on Ubiquitous Computing. ACM, New York, pp. 10–19 (2008)
[13] T. Huynh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In: 3rd International Symposium on Location- and Context-Awareness (LoCA), pp 50–67 (2007)
[14] H. Tian, P. Lei, L. Xingjuan, and X. Shusong. Wearable activity recognition for automatic microblog updates. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1720–1723 (2009)
[15] Android 2.2 Platform, http://developer.android.com/sdk/android-2.2-highlights.html
[16] D. Aha, D. Kibler, and M. K. Albert. Instance-Based Learning Algorithms. Machine Learning, vol. 6 pp. 37–66 (1991)
[17] A. Carlotto, M. Parodi, C. Bonamico, F. Lavagetto, and M. Valla. Proximity classification for mobile devices using wi-fi environment similarity. In: ACM International Workshop. (2008)
[18] P. C. Ribeiro, and J. Santos-Victor. Human activity recognition from video: Modeling, feature selection and classification architecture, In: Proc. Int. Workshop Human Activity Recognition and Modeling, pp. 61–70 (2005)