proceedings

MDPI

Proceedings

Multi-Event Naive Bayes Classifier for Activity Recognition in the UCAmI Cup [†]

Antonio R. Jiménez * and **Fernando Seco**

Centre for Automation and Robotics (CAR), Consejo Superior de Investigaciones Científicas (CSIC)-UPM, Ctra, Campo Real km 0.2, 28500 La Poveda, Arganda del Rey, Madrid, Spain; fernando.seco@csic.es

* Correspondence: antonio.jimenez@csic.es; Tel.: +34-918711900

[†] Presented at the 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2018), Punta Cana, Dominican Republic, 4–7 December 2018.

Published: 18 October 2018

Abstract: This short paper presents the activity recognition results obtained from the CAR-CSIC team for the UCAmI'18 Cup. We propose a multi-event naive Bayes classifier for estimating 24 different activities in real-time. We use all the sensorial information provided for the competition, i.e., binary sensors fixed to everyday objects, proximity BLE-based tags, location-aware smart floor sensing and the wrist's acceleration. The results using training data-sets of 7 days show accuracies (true positives) about 68%; however for the three extra data-sets of the competition we were able to reach a 60.5% accuracy.

Keywords: competition; activity recognition; naive bayes classifier; real-time classifier; bluetooth proximity; acceleration; binary sensors; capacitive floor

1. Introduction

Several activity recognition competitions have already been proposed during the last years. Some examples are CVPR and VISUM which are focused on the analysis of video and images to deduce the activities taking place [1,2]. Activity recognition (AR), is more than detecting actions in video frames, and it is a very challenging topic that has been studied by many research groups. The different approaches found in the literature differ mainly in terms of the used sensor technology, the machine learning algorithms and the realism of the environment under test [3]. Regarding sensors, apart from video, some works include the use of wearables such as smartwatches that include accelerometers or gyroscopes that allow the detection of activities that depend on the motion or orientation of the person (standing, lying on the bed, walking, etc.) [4]. More common is to use environment sensors such as infrared motion detectors (PIR) or reed switches coupled to doors or objects that must be placed on a base. With this kind of sensor is possible to detect when a person leaves or enters home, use the dishwasher or take a remote control [5]. Other environment sensors have been explored using RFID tags or BLE beacons for proximity detection.

Many different algorithmic approaches have been presented [3,5–10] trying to achieve good accuracies in activity recognition, from naive Bayes, hidden Markov classifiers, AdaBoost classifiers, Decision trees, Support vector machines or conditional random fields, all combined with different heuristics, windowing and segmentation methods. In short paper we will describe the use of a naive Bayes approach with emphasis on multi-type event-driven location-aware activity recognition. We will make use of all datasets available for the competition. We will not use any segmentation phase, so algorithms interpret the received sensor events as soon as they are measured and activity estimations are generated in real-time. The naive Bayes classifier is complemented with an activity prediction model that is used in order to guess the more-likely next activities to occur under a recursive Bayesian estimation approach.

Next Section 2 explains the methodology, and the following ones the modeling of activity sequences, the modeling of sensor events, and the final Section 5 the activity classification results.

2. Methodology

In the literature there are mainly three common approaches for processing streams of data [3]: (1) Explicit segmentation, (2) Time-based windowing and (3) Sensor event-based windowing. The explicit segmentation process tries to identify a window where an individual activity could be taking place, and the purpose is to separate (segment) those time intervals for a second classification stage. The second approach, the time-based windowing, divides the entire sequence of sensors events into smaller consecutive equal-size time intervals. On the other hand, the sensor event-based windowing divides the sequence into windows containing equal number of sensor events. The problem of all these approaches is defining the criteria to know how to select the optimal window values, or the number of events within a window. The result of the segmentation gives a sequence of non-overlapping intervals, so if the found intervals are too small or two large, then the classification can be confused since several activities could be present in one segment, or on the contrary, just a fraction of an activity could appear in the window.

We propose to use a fixed-size moving overlapping window to avoid doing an explicit data segmentation. We process the events as they are received, in real-time, but we do not assume that the time window contains an activity that must be classified. We assume that the window contain information that can be used to accumulate clues that increase the probability of being doing a particular activity. This segmentation-free approach is implemented using an iterative activity likelihood estimation while the fixed window is moved over time (at one-second interval displacements). A recursive Bayes filter is implemented as an improved version of a naive-Bayes classifier. Instead of doing an static classification based on the events present in a window, we do a dynamic process. The method uses an activity state vector $x = (w_1, w_2, \ldots, w_a, \ldots, w_n)$ representing the likelihood of doing a given activity a, where $a \in 1 \ldots n$, being $n = 24$ the number of different activities. The weights w_a of the activity state vector x evolves over time as new overlapped windows containing events are received. The Bayes filter approach allow the use of process models (probability of transition from an activity to a different one) and measurement models (probabilities of receiving an event for each activity). The final Bayes classifier is implemented using a decision rule (maximum a posteriori or MAP) that takes as the classified activity that with the maximum probability or weight in the activity vector.

3. Modeling Activity Sequences

Dataset were analyzed in order to see the number of occurrences, the mean duration of each activity, the minimum or maximum time and its percentage of change respect to the mean value (Δt). Table 1 shows this analytic results. A total of 169 activities are detected in those 7 days. A few high frequency activities (more than 7 times in 7 days) are detected, being: Brush teeth (21 times, i.e., 3 times a day), dressing (15 times), entering/leaving the smartlab (12/9 times), put waste in bin (11) and using the toilet (10). Unfrequent activities are playing a video game (1), relax on the sofa (1), visit (1), dishwasher (2) and work on a table (2).

We also analyzed the correlation between one activity type and the next one, in order to identify a repetitive sequence pattern. This analysis is presented in Figure 1. It can be seen that activities number 2, 3 and 4 are always followed by activities 5, 6 and 7 (i.e., after *Prepare breakfast* the next activity is *Breakfast*, after *Prepare lunch* the next activity is *Lunch*, and after *Prepare dinner* the next activity is *Dinner*). We observe that after activity 7 (Dinner) is quite probable to do activity 1 (Take medication).

Table 1. Statistics from the training set of activities (7 days).

Activities		Time in Seconds			Δt(%)	Count
ID	**Description**	**Mean**	**Max**	**Min**		
1	Take medication	100	158	60	98	7
2	Prepare breakfast	198	374	89	144	7
3	Prepare lunch	483	614	173	91	6
4	Prepare dinner	244	314	176	57	7
5	Breakfast	258	379	167	82	7
6	Lunch	401	649	303	86	6
7	Dinner	350	454	195	74	7
8	Eat a snack	67	95	46	74	5
9	Watch TV	420	792	123	159	6
10	Enter the SmartLab	51	73	38	68	12
11	Play a videogame	401	401	401	0	1
12	Relax on the sofa	1046	1046	1046	0	1
13	Leave the SmarLab	51	73	38	69	9
14	Visit in SmartLab	43	43	43	0	1
15	Put waste in the bin	136	247	73	128	11
16	Wash hands	57	67	36	54	6
17	Brush teeth	79	126	47	100	21
18	Use the toilet	49	70	36	69	10
19	Wash dishes	40	44	36	20	2
20	Wash clothes	54	63	45	33	6
21	Work at the table	512	637	387	49	2
22	Dressing	98	168	60	110	15
23	Go to the bed	90	128	54	82	7
24	Wake up	123	202	53	121	7
25	None	23	155	1	658	169

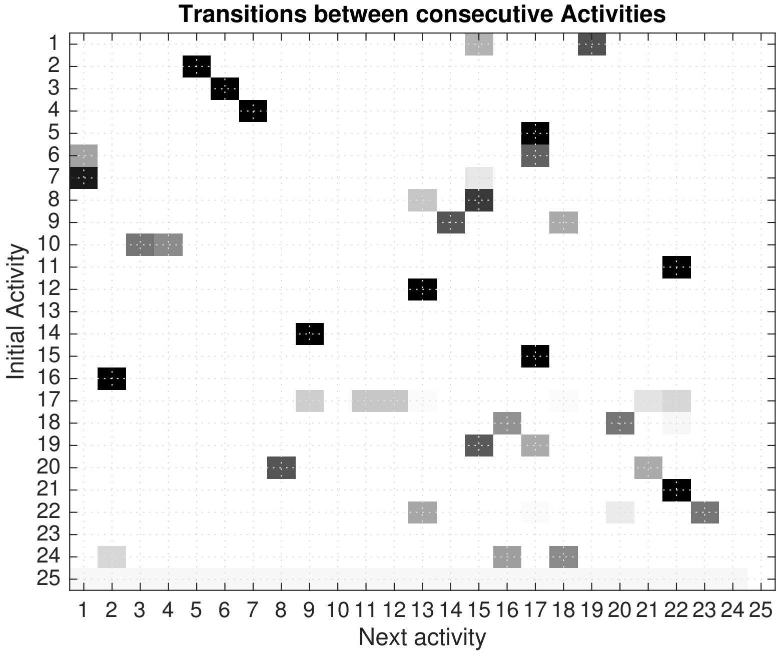

Figure 1. Transition matrix relating activities with the next activity, obtained from the seven days training set.

Many other activity transitions are correlated, and we can take advantage of this most probable activity propagation to forecast the next activity to come. Details of how we did it is presented in [11].

4. Modeling of Sensor Events

In this subsection we will show the different relations between the different sensor events and the performed activities. We will concentrate the description on binary sensors. The creation of the other measurement model relating activities and events (floor, proximity and acceleration) can be found in [11].

Observing the binary events for the whole seven-days training set, we obtained the probability relation matrix in Figure 2. We can observe that some sensor events clearly identify certain activities, for example, binary 5 (wardrobe clothes) is correlated with activity 22 (Dressing); or binary 19 (Fruit platter) is correlated with activity 8 (Eat a snack). On the contrary, some sensor events do not clearly relates to any activity; this is the case of most motion sensors (numbers 6, 7, 8, 9) which corresponds to presence detection at the kitchen, bed, bedroom and sofa, respectively.

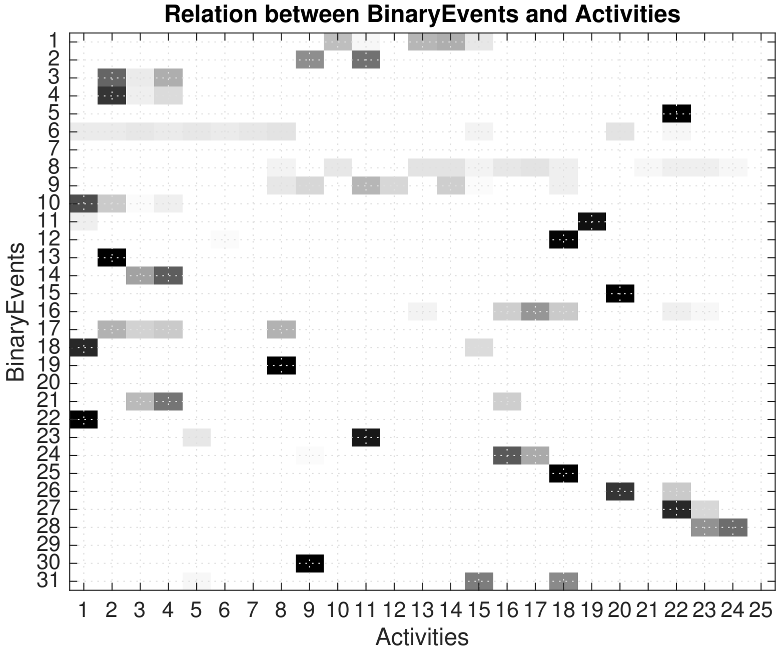

Figure 2. Measurement matrix relating binary events with activities, obtained from the seven days training set.

The activity clues derived from binary sensors are accumulated in an auxiliary state vector x_{binary} that is computed as follows:

$$x^k_{\text{binary}} = \sum_{b=1}^{31} \delta(b) * \text{binary}(b,:), \tag{1}$$

where $\delta(b)$ is the dirac function if a given binary b is found in the 90-s window. And $\text{binary}(b,:)$ is the binary relation vector extracted from one row out of 31 binary events in matrix in Figure 2.

Other information, apart from binary events were used, such as proximity events created when BLE RSSI is strong enough, a floor tile is stepped, or from other events created when the standard deviation of the acceleration magnitude is higher than a given threshold. The final measurement model integration of the whole weighs coming from sensor events (Binary, BLE, Floor and Acce) and time periods is done as follows:

$$x^k = \{w1 * x^k_{\text{Binary}} + w2 * x^k_{\text{BLE}} + w3 * x^k_{\text{Floor}} + w4 * x^k_{\text{Acce}}\} * x^k_{\text{Day}}, \tag{2}$$

where $w1$, $w2$, $w3$ and $w4$ are arbitrary weights to take more into account some sensor events than others. In the implementation used to generate the results shown in next section, we used these values, $(1, 0.5, 0.7, 0.3)$, respectively.

The reason for adding up the clues from sensors instead of multiplying them, as the naive principle of independent measurement suggest, is for increasing the robustness of sensor condition registration. In many situations not all sensor events are triggered so it could lead to many activities being rejected, when in reality they could be being performed, so causing frequent degeneration of the probability vector (all vector equal to zero in all their activities). The addition of clues, in a voting manner, makes the solution more robust against sensor noise or incomplete measurement models.

5. Activity Recognition Results

The overall detection results for the 7 days tests are shown in Figure 3 where a confusion matrix is presented. There is a predominant diagonal dark line, which represent the correct detections of activities, but also some off-diagonal estimations that represent estimation errors.

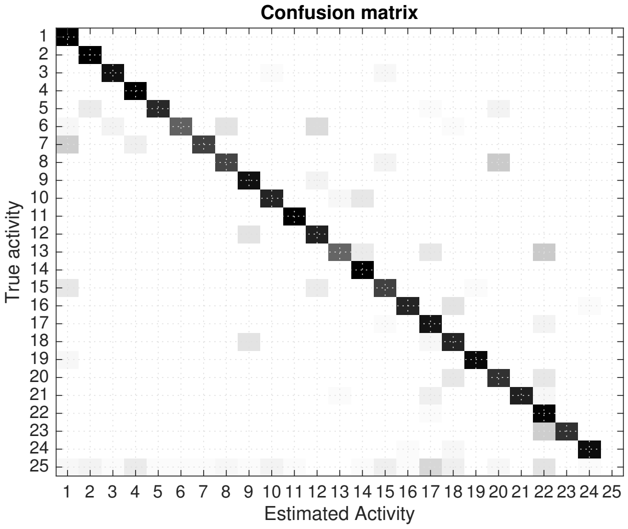

Figure 3. Confusion matrix.

The best performance (using as validation test the same logfiles used for trainnig) is an 83% of true positives (in-diagonal estimates). When using the competition dataset with three extra days, the performance dropped to a 60.5% percentage. These results are obtained activating all the different event sensor streams (*Binary*, *BLE*, *Floor*, *Acce*). We observed an accumulative increased performance when adding more sensor events together, being the *Acce* events the ones with less contribution. Taking into account, the large number of different activities (24) in the dataset, and the generality and simplified version of the algorithm, we believe that the results are good. As a future work we would like to compare these results with other using more sophisticated approaches (Random forest, SVM, etc.) with exactly the same dataset, in order to see the quality of the results.

6. Conclusions

In this short paper we have shown some partial details of the methods used to compete in the UCAmI Cup. We proposed to use a naive Bayes implementation with emphasis on multi-type event-driven location-aware activity recognition. Our method combined multiple events generated by binary sensors fixed to everyday objects, a capacitive smart floor, the received signal strength (RSS) from BLE beacons to a smart-watch and the sensed acceleration on the actor's wrist. The method did not used an explicit segmentation phase, it interprets the received events as soon as they are measured and activity estimations are generated in real-time without any post-processing or time-reversal re-estimation. An activity prediction model is used in order to guess the more-likely next activity to occur, and several measurement model are added-up in order to reinforce the believe in activities. A maximum a posteriori decision rule is used to infer the most probable activity. The evaluation results show an improved performance while adding new sensor type events to the activity engine estimator. Accuracy results within the competition was just a 60.5% of true positives, which is an acceptable figure taking into account the high number of different activities to classify (24 activities).

Acknowledgments: The authors thank UCAmI'18 Cup team and the financial support received from projects: TARSIUS (TIN2015-71564-C4-2-R (MINECO/FEDER)) and Tomoloc (CSIC-PIE Ref. 201450E011).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. VISUM Activity Recognition. 2018. Available online: https://www.kaggle.com/c/visum-activity-recognition#description (accessed on 30 May 2018).
2. CVPR Activity Recognition Challenges. 2018. Available online: http://www.umiacs.umd.edu/conferences/cvpr2011/ARC/ (accessed on 30 May 2018).
3. Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **2014**, *10*, 138–154, doi:10.1016/j.pmcj.2012.07.003.
4. Kwon, Y.; Kang, K.; Bae, C. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Syst. Appl.* **2014**, *41*, 6067–6074. doi:10.1016/j.eswa.2014.04.037.
5. Espinilla, M.; Medina, J.; Calzada, A.; Liu, J.; Martínez, L.; Nugent, C. Optimizing the configuration of an heterogeneous architecture of sensors for activity recognition, using the extended belief rule-based inference methodology. *Microprocess. Microsyst.* **2017**, *52*, 381–390, doi:10.1016/j.micpro.2016.10.007.
6. Fang, H.; He, L.; Si, H.; Liu, P.; Xie, X. Human activity recognition based on feature selection in smart home using back-propagation algorithm. *ISA Trans.* **2014**, *53*, 1629–1638, doi:10.1016/j.isatra.2014.06.008.
7. Gravina, R.; Alinia, P.; Ghasemzadeh, H.; Fortino, G. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Inf. Fusion* **2017**, *35*, 1339–1351, doi:10.1016/j.inffus.2016.09.005.
8. Hong, X.; Nugent, C.; Mulvenna, M.; McClean, S.; Scotney, B.; Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive Mob. Comput.* **2009**, *5*, 236–252. doi:10.1016/j.pmcj.2008.05.002.
9. Kim, E.H.; Oh, S.K.; Pedrycz, W. Reinforced rule-based fuzzy models: Design and analysis. *Knowl. Based Syst.* **2017**, *119*, 44–58, doi:10.1016/j.knosys.2016.12.003.
10. Minor, B.; Doppa, J.R.; Cook, D.J. Data-Driven Activity Prediction. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining–KDD '15, Sydney, Australia, 10–13 August 2015; pp. 805–814, doi:10.1145/2783258.2783408.
11. Jimenez, A.R.; Seco, F. Event-driven Real-time Location-aware Activity Recognition in AAL Scenarios. In Proceedings of the 2th International Conference on Ubiquitous Computing and Ambient Intelligence–UCAmI, Punta Cana, Dominican Republic, 4–7 December 2018; pp. 1–12.