Activity Recognition Using Dynamic Instance Activation

Alberto Calzada[1], Chris Nugent[1(✉)], Macarena Espinilla[2], Jonathan Synnott[1], and Luis Martinez[2]

[1] School of Computing and Mathematics, University of Ulster, Jordanstown, Northern Ireland, UK
{a.calzada,cd.nugent,j.synnott}@ulster.ac.uk
[2] Department of Computer Sciences, University of Jaen, Jaen, Spain
{mestevez,martin}@ujaen.es

Abstract. Dynamic Instance Activation (DIA) is a newly developed data-driven classification algorithm. It was designed to minimise the negative impact in situations of data incompleteness and inconsistency. To achieve this, the proposed methodology attempts to maximise the accuracy of the classification process in a way that does not compromise the overall computational effort.

In this research, DIA was evaluated in the context of Human Activity Recognition (HAR) for Smart Environments, using datasets consisting of binary sensor data and their associated class labels (activities). This scenario was selected as an ideal case study to illustrate the usefulness of DIA considering the wide range of domains in which HAR is applied. It was also considered adequate given the simplicity of the data involved in the process, which allows using relatively simple similarity functions, therefore placing the main focus on DIA's performance.

In this context, DIA was compared with other state-of-the-art classifiers, delivering promising results in terms of percentage of activities correctly identified over the total. It is important to note that these results could be further improved if other similarity functions or data representation schemes were selected.

Keywords: Activity recognition · Classification · Smart environments · Similarity measures

1 Introduction

Over the last decade, Human Activity Recognition (HAR) has emerged as a prominent field of study from which several research areas have directly benefited. These areas include pervasive, mobile and context-aware computing, surveillance and ambient-assisted living, to name but a few. This increased attention in the study of HAR has been driven by two main aspects: technological advances in sensor devices and application-domain demands [1]. On the one hand, innovations in miniaturization, wired and wireless communications, reduced costs, decrease in energy consumption and increased capacity have increased the use of sensing devices.

C.R. García et al. (Eds.): UCAmI 2016, Part II, LNCS 10070, pp. 75–83, 2016.
DOI: 10.1007/978-3-319-48799-1_9

One particular application domain that has greatly benefited from the advances in sensor devices is that of Smart Environments, which often require extensive sensor-based monitoring across a particular scenario [1]. Among all the types of Smart Environments, the scenarios consisting solely of binary sensors can be considered as one of the simplest. In this regard, Binary Smart Environments (BSEs) provide an ideal framework in which to test new concepts and procedures that can be applied in the entire HAR field. The research presented in this paper can be considered as an example of research within this BSE domain.

The activities performed in BSE scenarios in a certain period of time are described as sequences of one or more sensors. The status of a sensor is represented as a binary variable that takes the value 1 if the sensor was activated and 0 otherwise. When these sequences of sensor statuses (state sequences) and their associated class label (the activity performed when the sequence of sensor activations occurred) are collected within a dataset, a data-driven machine learning (ML) algorithm can be used to classify new sequences of events and identify the activity being performed [1, 2].

In this regard, HAR can be described as a classification problem. As such, it can be negatively impacted by situations of incompleteness and inconsistency of instances (in this case, sequences of binary sensor state sequences). On the one hand, a situation of incompleteness occurs when the data does not contain an exact match for a new instance that needs to be classified, and therefore a new activity cannot be identified. On the other hand, a situation of incompleteness arises when a group of similar instances included in the dataset are labelled with different activities. In this case, when a new instance that needs to be classified is assessed, the classification algorithm is prone to misclassify the new activity.

Situations of incompleteness and inconsistency can be considered as opposite: while the former occurs due to a lack of data, the latter can originate due to an excess of data. The proposed Dynamic Instance Activation (DIA) approach in the current work balances these two situations by finding a non-empty set of instances similar to the one being classified and with minimum inconsistency between them [2]. Hence, only the most relevant and consistent observations are then aggregated, therefore producing a more clear output. DIA can be considered as a generalization of the recently-proposed Dynamic Rule Activation (DRA) algorithm [3], which was designed to work solely on Extended Belief Rule-Based systems (E-BRB) [4]. DIA removes the need for rule-based datasets that DRA had, and is able to operate over any type of dataset, provided that a similarity and an aggregation function are provided.

The following Section illustrates the concepts of incompleteness and inconsistency of information in a BSE and Sect. 3 details the DIA process. Section 4 briefly introduces several BSE case studies where DIA and other popular ML classifiers were applied and in Sect. 5 their results are highlighted. Finally, the conclusions are outlined in Sect. 6.

2 Data Incompleteness and Inconsistency

In a BSE, incomplete and inconsistent situations are likely to occur due to the nature of the data and the elements related to its generation. For example, a human might perform

the different tasks of an activity in a different order, timescale or miss an optional task. Also, a sensor device might be faulty, out of battery, misplaced or not connected to the network. Any of these situations introduce some noise in the dataset being compiled which affects the quality of the data. Therefore, an HAR algorithm should be designed to acknowledge these situations and have the capability to minimise their impact in the activity recognition process. Two of the undesirable data issues commonly encountered are incompleteness and inconsistency, which are briefly detailed in an BSE context in the following Sections.

2.1 Incompleteness

In a BSE, a situation of incompleteness occurs when the sensor state sequence analyzed to determine the activity performed does not exactly match any instance of the BSE training dataset. This can be due to some noise which was induced in the data as previously mentioned, or due to the activity which was being performed having not been recorded previously.

For example, in a basic BSE consisting of two sensors placed in a bathroom: one of them activates when the door is open (Bathroom Door Sensor) and another one when the toilet is flushed (Toilet Sensor). Consider the dataset presented in Table 1, consisting of three instances.

Table 1. Example of a basic BSE dataset

Instance number	Bathroom door sensor status	Toilet sensor status	Activity
1	1	1	Use toilet
2	1	0	Use toilet
3	1	1	Clean toilet

Considering the BSE dataset illustrated in Table 1, and the sensor state sequence: *{(Bathroom Door Sensor,* 0*), (Toilet Sensor,* 1*)}*. Since such a sequence is not represented in the BSE dataset, which is being used for training a HAR algorithm, will not be able to exactly match any instance in order to confidently identify an activity based on the examples. Hence, the BSE dataset can be considered as incomplete.

2.2 Inconsistency

A situation of inconsistency occurs when the sensor state sequence of an activity needs to be recognized matches two or more different activities recorded in the dataset. In such a situation, a HAR algorithm would typically be unable to select between the range of matched activities.

In the example represented in Table 1, the sensor state sequence *{(Bathroom Door Sensor,* 1*), (Toilet Sensor,* 1*)}* matches two instances labelled with different activities (*Use Toilet* and *Clean Toilet*). These two instances are said to be inconsistent between them, and it will be difficult for a HAR algorithm to discern which one of the two activities was the one performed.

The following Section describes the DIA algorithm, which aims to address these two situations simultaneously, subsequently minimizing their impact.

3 Dynamic Instance Activation

In many classification algorithms, two differentiated steps can be identified: similarity evaluation and aggregation. In the first step, the similarities between the instances in the training dataset and the unlabelled instance (the one being classified) are measured in order to select the most relevant instances from the dataset. In the second step, these relevant instances and their similarities are aggregated in a certain way in order to assign a label to the unlabelled instance. In terms of HAR, this process is used to assign an activity to the unlabelled instance (i.e. sensor state sequence).

The DIA process sits between these two steps. Its main aim is to minimize the incompleteness and inconsistency in the set of instances obtained after the similarity evaluation step. Therefore, a more suitable set of instances is aggregated in the subsequent step, eventually increasing the accuracy of the classifier.

As illustrated in Fig. 1, DIA focuses on the set of relevant instances obtained from the similarity step. It increases the size of this set if incompleteness is found, and reduces it when a situation of inconsistency occurs. This is how DIA attempts to minimize incompleteness and inconsistency simultaneously. Once a suitable size of relevant instances is identified, it can be processed in the final aggregation step.

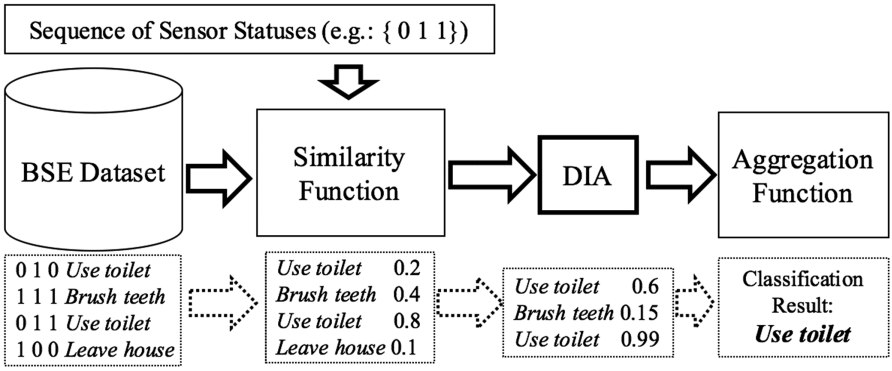

Fig. 1. The function of DIA embedded in a classification process. Example elements added for clarification are contained within dotted rectangles.

The increase and reduction process in which DIA is based is achieved by optimizing one single parameter, named λ. This parameter modifies the original similarities provided by a similarity function as follows:

$$S_\lambda(X, I_i) = (S(X, I_i))^\lambda \tag{1}$$

where X is the unlabelled sensor state sequence being classified, and I_i is the sensor state sequence of the i^{th} instance of the BSE dataset. S is a similarity function that retrieves values within the [0, 1] range when comparing two sensor state sequences, where 0 and 1 denote total dissimilarity and exact match, respectively.

By augmenting the value of λ, the set of relevant instances is reduced as an attempt to minimize inconsistency. Likewise, as the value of λ is decreased, the set of instances is enlarged, reducing the effects of incompleteness. For each λ value, an agreement function is necessary to compare the sets produced against other λ values. In this research, the agreement function is defined as the highest percentage of instances whose label is the same over the total set of instances contained in the set. Once a series of λ values is evaluated, the set of instances with the highest agreement is passed onto the aggregation step.

DIA allows the modification of the similarity and aggregation functions of the classifier where it is being used. Nevertheless, for Eq. 1 to function correctly, the selected similarity function S needs to return values within the interval [0, 1], as previously described.

4 Case Studies

The Hamming distance [5, 6], was selected as the similarity function since it is designed to compare different sequences of binary values. Therefore, it is ideal in order to compare sensor state sequences for different activities stored in a BSE.

In addition, in order to place the main focus on DIA, the aggregation function selected was the average of the similarity values (after being tuned by DIA) per each activity included in the set of instances retrieved by DIA. The activity with the highest average is chosen as the classification result. Average was selected as it is probably one of the simplest aggregation functions to work on.

This simple DIA-focused classifier was called Dynamic Instance Activation with Hamming and Average (DIAHA). Hence, its equivalent without using DIA can be named HA.

A total of 4 datasets collected from distinct BSEs were inputted into DIA in order to measure its accuracy, in terms of percentage of instances correctly predicted.

The first activity recognition dataset tested has been widely used in research publications and it was presented in [7]. It was compiled in a house environment consisting of three rooms in which a male participant performed 7 types of ADLs during 25 days. The information about each one of the 245 observations recorded was generated using 14 binary sensors placed across the home setting.

The second dataset (WSU) utilized in this study was collected from the smart apartment test-bed of the Washington State University (WSU) by 24 participants who completed five activities [8]. These activities consisted of the following: Make a phone call, Wash hands, Cook, Eat and Clean. Activities were annotated by the participants while performing each activity. In this paper, the original dataset was transformed into a set of activity binary vectors where each vector was an activity and each component of the vector represented the sensor state sequence during the performance of the activity.

As for the rest of the datasets generated from BSEs, a sensor status with a value of one indicates that the sensor was activated at least once, while a value of zero indicates that the sensor was not activated. The number of sensors was 39, and can be categorized as: M01… M26: motion sensor, I01: item sensors for oatmeal, I02: raisins, I03: brown sugar, I04: bowl, I05: measuring spoon, I06: medicine container sensor, I07: pot sensor, I08: phone book sensor, D01: cabinet sensor, AD1-A: water sensor, AD1-B: water sensor, AD1-C: burner sensor, asterisk: phone usage.

The remaining two datasets were generated thanks to the Open Data Initiative (ODI) for Activity Recognition consortium, which currently consists of researchers from Ulster University (UK), Luleå Technical University (Sweden), Halmstad University (Sweden), University of Jaén (Spain), and the University of Twente (The Netherlands) [9]. The consortium was recently created recognising that the development of novel activity recognition approaches requires access to high quality, annotated datasets capturing activity performance through sensor data. Large amounts of efforts have been put in place internationally for the designing of experiments, collecting and analyzing data. Nevertheless, the size, diversity and availability of datasets is limited due to significant financial and time costs associated with running experiments within intelligent environments. There is a general appreciation that further efforts should be made to streamline existing approaches to define common data formats, data collection protocols, data aggregation platforms and approaches for comparison of analysis techniques. The ODI has been established to address this, by aiming to create a structured approach to provide annotated datasets in an accessible format.

The first ODI dataset (ODI_1), consisted of sensor data generated by 4 participants within the IE Sim intelligent environment simulation tool [10]. Each participant completed 11 activities 7 times (having a total of 308 observations). These activities consisted of the following typical household activities: Go to Bed, Use toilet, Watch Television, Prepare Breakfast, Take Shower, Leave House, Get Cold Drink, Get Hot Drink, Prepare Dinner, Get Dressed and Use Telephone. These activities were annotated by the participants, while performing them. The dataset was provided in 2 formats: MySQL database entry and activity vector. The latter version is the one being utilized in this study, given that it matches the format of the datasets previously presented. The second dataset generated by the ODI (ODI_2) consists of the previous dataset with an additional 4 participants performing the same 11 activities 7 times each (having a total of 616 observations). Using these four datasets, DIAHA, HA and four popular state-of-the-art classifiers were tested running a series of 50 10-fold cross-validation (CV) tests. In each test of the series of 50, the observations in each dataset were randomly permuted. In this way, the CV tests in each series are based on different training/testing instances.

A selection of some of the most popular classifiers used in literature was used to compare the performance of HA with and without the DIA boost. The selected classifiers were: Naive Bayes classifier (NB) [11], Nearest Neighbor (NN) [12], Decision Table (DT) [13] and Support Vector Machines (SVM) [14].

5 Results

After running the series of tests specified in Sect. 4, the accuracy of each method, measured as the percentage of activities correctly classified over the total, was used to measure the performance of each classification method used for HAR. Table 2 summarizes the results obtained from the tests, with the results of the two best classifiers in each dataset highlighted in bold.

Table 2. Average accuracy results obtained from the 50 10-fold cross-validation tests for all the classifiers and datasets evaluated.

Method	Dataset				Mean
	Van Kasteren	WSU	ODI_1	ODI_2	
HA	46.53 %	55.03 %	70.04 %	79.72 %	62.83 %
DIAHA	**96.26 %**	**94.51 %**	**95.38 %**	93.28 %	**94.85 %**
DT	95.19 %	88.22 %	95.24 %	93.26 %	92.08 %
NN	95.94 %	93.40 %	95.01 %	91.65 %	94.00 %
NB	**96.78 %**	**94.18 %**	**96.37 %**	**94.04 %**	**95.34 %**
SVM	96.05 %	93.08 %	95.36 %	**94.55 %**	94.76 %

As illustrated in Table 2, when the HA method is complemented with the DIA enhancement, its results considerably increase. Note that, even if HA initially produces considerably worse results than the rest of classifiers, its performance is brought into competitive levels when it is boosted with DIA. This difference in performance helps quantifying the amount of incompleteness and inconsistency that the DIA boost is able to eliminate from a classifier (in this case: HA). In fact, this is the same reason why DIAHA is slightly better than NN on every occasion: DIAHA is able to detect when a nearest neighbor is an inconsistent sample, so it always looks to other surrounding neighbors for a greater agreement level. Such an agreement level is measured by an agreement function, as explained in Sect. 4.

One of the reasons why DIAHA outperforms DT and SVM in most cases is because DIA does not produce boundaries within the data set. Therefore, it does not classify based on the region in which a new observation falls, like DT and SVM do [13, 14] (although in different ways). On the contrary, DIA can be considered as an NN-based booster algorithm that dynamically selects the optimal number of neighbors to be considered when classifying each new observation.

A possible reason why DIAHA was slightly outperformed by NB in many cases is because NB considers the class probabilities in addition to the distribution of the values for each feature when producing a classification probability [12]. This suggests that one of the ways to improve DIA is to assume either or both of these factors in the selection of the λ parameter. A possible way of doing so is by use of either of those probabilities/distributions to weight the relevant samples obtained for each λ value and produce a distribution of class probabilities as a result; as opposed to the single class value being produced at the moment. Another possibility to improve the results provided by this research is to use a base classifier slightly more complex than HA. In this situation, DIA

would have better *initial predictions* to optimise/boost, instead of those rather low accuracies provided by HA.

Taking these results into consideration, the DIA booster will outperform other conventional classification methods in situations where the classes of the dataset (in this case activities to be recognised) cannot be separated in regions, taking their features as coordinates. This is the reason why DIAHA was able to improve the results of both DT and SVM. However, it was just slightly better than NN. In this regard, DIA is ideal in situations where NN already performs well, however, it still misclassifies observations due to high inconsistency levels.

In summary, it can be considered that the optimal situation for use of DIA is when similar instances have associated different classes and the dataset classes are not easily divided into regions by one or more features (in our case, binary sensor values).

It is important to note, however, that despite the simplicity of both the distance and aggregation functions utilized in HA, the DIAHA algorithm performs better than any other classifier in the WSU dataset. In addition, it obtains the second best result in the van Kasteren and ODI_1 dataset and it ranks second when the results from all the datasets are averaged, only behind the NB method.

6 Conclusion

This research demonstrated the value of the DIA method as an algorithm able to outperform the performance of other classifiers based on similarity and aggregation functions. Such an enhancement is based on identifying and minimising situations of incompleteness and/or inconsistency in the data being considered.

In order to illustrate these capabilities, the field of HAR based on binary sensors was selected as an ideal scenario where the data collected from different sensor devices is both as simple as possible and prone to be noisy. In such scenarios DIA could be easily exemplified, as found in the case studies when the performances of HA and DIA-HA were compared. Although a very simple classifier based on the Hamming distance similarity and a Weighted Average aggregation was proposed (HA), DIA has a great potential to be further improved if other more advanced similarity and aggregation functions were utilized.

Further research will therefore be focused on developing similarity and aggregation functions specialised on specific features of HAR, such as imbalanced classification or missing values. The development of algorithms able to include data from different types of sensors, rather than just from binary sensors, is also an aim to be addressed in the foreseeable future.

Acknowledgements. Invest Northern Ireland partially supported this project under the Competence Centre Program Grant RD0513853 – Connected Health Innovation Centre.

References

1. Chen, L., Hoey, J., Nugent, C.D., Cook, D.J., Yu, Z.W.: Sensor-based activity recognition. IEEE Trans. Syst. Man Cybern. Part C **42**(6), 790–808 (2012)
2. Calzada, A., Nugent, C., Espinilla, M., Martinez, L.: Generalized dynamic instance activation for activity recognition. In: 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Orlando, Florida, US, 17–20 August 2016
3. Calzada, A., Liu, J., Wang, H., Kashyap, A.: A new dynamic rule activation method for extended belief rule-bases. IEEE Trans. Knowl. Data Eng. **27**(4), 880–894 (2014)
4. Liu, J., Martinez, L., Calzada, A., Wang, H.: A novel belief rule base representation, generation and its inference methodology. Knowl. Based Syst. **53**, 129–141 (2013)
5. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
6. Norouzi, M., Fleet, D.J., Salakhutdinov R.R.: Hamming distance metric learning. In: Advances in Neural Information Processing Systems, pp. 1061–1069 (2012)
7. van Kasteren, T.L.M., Englebienne, G., Kröse, B.J.A.: Transferring knowledge of activity recognition across sensor networks. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive 2010. LNCS, vol. 6030, pp. 283–300. Springer, Heidelberg (2010)
8. Cook, D., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. Methods Inf. Med. **48**(5), 480–485 (2009)
9. Nugent, C., Synnott, J., Santanna, A., Espinilla, M., Cleland, I., Banos, O., Lundström, J., Hallberg, J., Calzada, A.: An initiative for the creation of open datasets within the pervasive healthcare. In: 10th EAI International Conference on Pervasive Computing Technologies for Healthcare, Cancun, Mexico, 16–19 May 2016
10. Synnott, J., Chen, L., Nugent, C.D., Moore, G.: The creation of simulated activity datasets using a graphical intelligent environment simulation tool. In: 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Chicago, IL, USA, 26–30 August 2014
11. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. **29**, 103–137 (1997)
12. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
13. Kohavi, R.: The power of decision tables. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 174–189. Springer, Heidelberg (1995)
14. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**, 273–297 (1995)