

Recognition of Activities in Resource Constrained Environments; Reducing the Computational Complexity

M. Espinilla¹(✉), A. Rivera¹, M.D. Pérez-Godoy¹, J. Medina¹, L. Martínez¹,
and C. Nugent²

¹ Computer Sciences Department, University of Jaén,
Campus Las Lagunillas S/n, 23071 Jaén, Spain

{mestevez,arivera,lperez,jmquero,martin}@ujaen.es

² School of Computing and Mathematics, University of Ulster,
Jordanstown BT37 0QB, UK
cd.nugent@ulster.ac.uk

Abstract. In our current work we propose a strategy to reduce the vast amounts of data produced within smart environments for sensor-based activity recognition through usage of the nearest neighbor (NN) approach. This approach has a number of disadvantages when deployed in resource constrained environments due to its high storage requirements and computational complexity. These requirements are closely related to the size of the data used as input to NN. A wide range of prototype generation (PG) algorithms, which are designed for use with the NN approach, have been proposed in the literature to reduce the size of the data set. In this work, we investigate the use of PG algorithms and their effect on binary sensor-based activity recognition when using a NN approach. To identify the most suitable PG algorithm four datasets were used consisting of binary sensor data and their associated class activities. The results obtained demonstrated the potential of three PG algorithms for sensor-based activity recognition that reduced the computational complexity by up to 95 % with an overall accuracy higher than 90 %.

Keywords: Activity recognition · Resource constrained environments · Nearest Neighbor (NN) · Prototype generation (PG) · Computational complexity

1 Introduction

Sensor-based activity recognition [1] is at the core of smart environments. This recognition aims to recognize the actions of one or more persons within the environment based on a series of observations of sub-actions and environmental conditions over a period of finite time. It can be deemed as a complex process that involves the following steps: (i) select and deploy the appropriate sensors to be attached to objects within the smart environment; (ii) collect, store and

pre-process the sensor related data and, finally, (iii) to classify activities from the sensor data through the use of activity models.

Advances in technology developments have mainly focused around the provision of a wide range of low cost devices, with low-power requirements and decreased form factor that implies constrained resource environments. An example are the inexpensive micro boards, such as Raspberry Pi [2] or Arduino [3]. These devices allow information to be read from sensors, in addition to the processing of sensor data to extract relevant information from the smart environments. These small computer boards, however, offer limited processing capacity and low storage capacity hence a key factor with their usage is to reduce the computational complexity of any tasks they must undertake [4, 5].

Approaches used for sensor-based activity recognition have been divided into two main categories: Data-Driven (DDA) and Knowledge-Driven (KDA) Approaches. The former, DDA, are based on machine learning techniques in which a preexistent dataset of user behaviors is required (and available). A training process is carried out, usually, to build an activity model which is followed by a testing processes to evaluate the generalization of the model in classifying unseen activities [6–8]. With KDA, an activity model is built through the incorporation of rich prior domain knowledge gleaned from the application domain, using knowledge engineering and knowledge management techniques [9, 10].

This contribution is focused on the most popular algorithm among all of the DDA solutions namely, the nearest neighbour (NN) [11] that provides simplicity and overall good levels of accuracy [12]. The approach is based on the concept of similarity patterns that can be allocated to the same class label (activity) [13–15].

The NN approach does, however, suffer from several shortcomings. These mainly relate to high storage requirements and high levels of computational complexity [16]. These requirements are closely related to the size of the dataset. The computational complexity of the linear search method of NN is $O(n d)$ where n is the size of the dataset and d is the dimensionality, i.e., the number of sensors. This fact is even more relevant in the application domain of activity recognition where the size of the data is related to a vast amount of generated-sensor data within smart environments.

For this reason, the PG algorithms [17] are a suitable choice given their focus on recognizing an optimal subset of representative samples from the original training data. This is achieved by removing noisy and redundant examples in order to generate and replace the original data with new artificial data [18].

In order to maximise the advantages provided by the NN approach and avoiding the drawbacks associated with the size of the datasets in resource constrained environments, the contribution of the current work proposes to use PG algorithms to reduce the size of the data in order to (i) decrease the storage requirements and computational complexity with the NN approach, (ii) maintain classification accuracy.

An evaluation is undertaken with four datasets to consider the effects of the reduction of the computational complexity in terms of overall accuracy for activity recognition based on sensor data gleaned from smart environments.

The remainder of the paper is structured as follows: Sect. 2 reviews the NN approach in addition to PG algorithms. Section 3 presents an empirical study that analyzes PG algorithms in terms of their accuracy and reduction for the purpose of activity recognition based on four datasets using binary sensors within smart environments. Finally, in Sect. 4, Conclusions and Future Work are presented.

2 Prototype Generation Algorithms Designed for the NN Approach

In this Section, we present an overview of the NN approach in addition to consider the notion of PG algorithms.

2.1 Nearest Neighbor Approach

The NN approach [11] is one of the most successfully used techniques for classification and pattern recognition tasks. It is based on the concept of similarity [19] and the fact that patterns that are similar, usually, have the same class label. The method is categorized as lazy learning [20] given it classifies the class label from raw training samples.

In order to recognize an unseen sample, representative training samples are stored within the activity model. Each training sample, which is annotated with a class label, is essentially a vector in a multidimensional feature space. In the case of activity recognition, each feature corresponds to each sensor of the network. During the testing process, a non-annotated vector i.e., a new sample, is classified. To do so, a parameter k is fixed that means the k training samples nearest to the new sample will be used to classify. The non-annotated vector, i.e., the new sample, is classified with the activity label corresponding to the most frequent label among the k training samples nearest to it. When $k = 1$ in the NN approach, i.e., the activity label of the non-annotated vector is the activity label of its single most closest neighbour.

The NN approach is based on the similarity of its k closest neighbours and has the ability to attain good levels of performance [12], however it suffers from the following three weaknesses [16]:

1. High storage requirements in order to retain the set of training samples.
2. High computational complexity in order to search through the training samples and classify a new sample.
3. Low tolerance to noise given that it considers all data relevant, even when the training set may contain incorrect data.

A successful technique, which has been shown to address these challenges, is based on PG algorithms. The following Section provides further details on PG algorithms and their use with NN approach.

2.2 Prototype Generation Algorithms

PG algorithms are a form of data reduction technique [21] that aim to identify an optimal subset from the original training set, by discarding noisy and redundant examples and by modifying the value of some features of the samples to build new artificial samples that are known as prototypes [18].

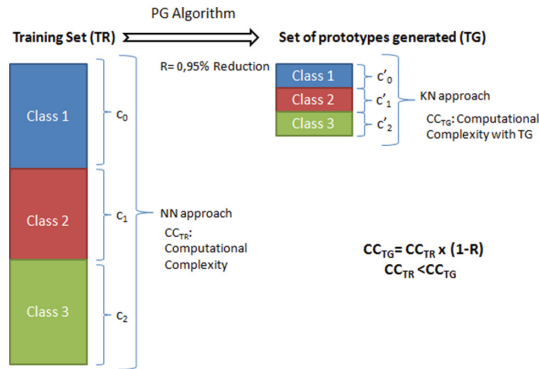


Fig. 1. Reduction through usage of PG algorithms in the number of stored instances with the ability to reduce the computational complexity of the NN

PG algorithms are therefore designed to obtain a set of prototypes generated TG , which has a smaller size of the data than the original training set TR . The cardinality of the TG is sufficiently small and has the subsequent effect to reduce both the storage requirements and computational complexity spent by the NN approach.

A wide range of PG algorithms have been designed for the NN approach to reduce the size of the dataset. Figure 1 illustrates the objective of these algorithms which have been categorized into a taxonomy based on the following four mechanisms of prototyping [17]:

- *Positioning adjustment* [22–24]: This technique corrects the position of a subset of prototypes from the initial set by using an optimization procedure. New positions of the prototype can be obtained by using the movement idea in the multidimensional feature space by adding or subtracting some quantities to the feature values of the prototypes.
- *Class relabeling* [25, 26]: This generation mechanism consists of changing the class labels of samples from TR , which are considered as having errors and/or belonging to other different classes than to those which they have been labeled.
- *Centroid based* [27, 28]: These techniques are based on generating artificial prototypes by merging a set of similar examples. The merging process is usually made from the computation of averaged attribute values over a selected set, yielding the so-called centroids.

- *Space splitting* [29,30]: These techniques are based on different heuristics to partition the feature space, along with several mechanisms to define new prototypes. The idea consists of dividing TR into regions, which will be replaced with representative examples establishing the decision boundaries associated with the original TR .

The PG algorithms can be associated with four types of reduction [17].

- *Incremental*: An incremental reduction starts with an empty reduced set TG or with only some representative prototypes from each class.
- *Decremental*: The decremental reduction begins with $TG = TR$, and then the algorithm starts reducing TG or modifying the prototypes in TG .
- *Fixed*. The fixed reduction establishes the final number of prototypes for TG using a user previously defined parameter related to the percentage of retention of TR .
- *Mixed*. A mixed reduction starts with a pre-selected subset TG , and following this, additions, modifications and removals of prototypes are performed in TG .

3 Case Study

This Section details the evaluations under taken to investigate the effects of the performance of the PG algorithms to decrease the size of the dataset for use with NN as a means of classification in the process of activity recognition.

3.1 Activity Recognition Datasets

The case study presented in this contribution uses four datasets collected from multiple smart environments all of which used binary sensors.

Each instance of the dataset is a vector with $d + 1$ components; the first d components correspond to the value of the d sensors involved in the smart environment and the last component, $d+1$ corresponds to the activity performed (class label). The value of a sensor is represented as a binary variable that takes the value 1 if the sensor had a change of state and 0 otherwise.

Following, the four datasets are described:

- *Casas* [31]. This dataset was collected from the smart apartment test-bed of the Washington State University that contains 121 instances that was generated using 39 binary sensors with five types of activities.
- *ODI1 and ODI2* [32].¹ *ODI* datasets were generated within the IE Sim intelligent environment simulation tool. The first *ODI* contains 308 observations generated using 21 binary sensors with 11 types of activities. The second *ODI* dataset contains 616 observations that was generated using also 21 binary sensors with the same 11 types of activities that *ODI1*

¹ *ODI1* and *ODI2* datasets were generated by Ulster University thanks to the Open Data Initiative (ODI) [33] for Activity Recognition consortium that aims to create a structured approach to provide annotated datasets in an accessible format.

- *VanKasteren* [34]. This dataset was compiled in a house environment and contains 245 observations that were generated using 14 binary sensors with seven types of activities.

Depending on each person within the smart environment, it is usual that the same activity may be performed in a number of different ways over a range of durations. Thus, depending on the activity performance the sensors' interactions can be different.

3.2 Evaluated PG Algorithms

Twelve PG algorithms, which are presented in see Table 1, have been considered in order to identify the most suitable PG algorithms for binary sensor-based activity recognition that reduce the computational complexity.

Table 1. Taxonomy of PG algorithms evaluated

PG algorithm	Mechanisms of prototyping	Reduction type
LVQ3	Positioning Adjustment	Fixed
VQ	Positioning Adjustment	Fixed
LVQTC	Positioning Adjustment	Fixed
MSE	Positioning Adjustment	Mixed
HYB	Positioning Adjustment	Mixed
LVQPRU	Positioning Adjustment	Mixed
PSCSA	Positioning Adjustment	Mixed
ENPC	Positioning Adjustment	Mixed
GENN	Class Re-labeling	–
MixtGauss	Centroids	Fixed
MCA	Centroids	Decremental
POC	Space Splitting	Incremental

Each PG algorithm requires a set of parameters. In this contribution, the fixed configuration has been the configuration proposed in [17] due to the successful results previously achieved.

Given that activities of the four datasets are annotated, we are able to evaluate the accuracy of the resulting subset of prototypes from each PG algorithm. In this way, the classification percentage is related to the accuracy percentage using the complete dataset to evaluate the NN classifier for a given prototype in the activity recognition process. Specifically, the accuracy percentage is defined as the proportion of true results among the total number of classes examined.

The set of evaluated PG algorithms are presented in Table 1, which were run using Keel software [35], an open source Java software tool with evolutionary

learning and soft computing based techniques for different kinds of data mining problems. To assess the performance of the PG algorithms, a 10-fold Cross-Validation was used to evaluate the accuracy percentage of each PG algorithm. The main advantage of this validation is that all the samples in the dataset are eventually used for both training and testing [36]. With this approach less of an emphasis is placed on how much the data becomes divided.

3.3 Results

Table 2 presents the average results obtained by the set of PG algorithms evaluated over the four datasets with the NN approach with $k = 1$ ($1NN$). The value of the parameter $k = 1$ is selected because this value presents low tolerance to noise in the NN approach.

Table 2 indicates the accuracy percentage in addition to the percentage reduction in terms of computational complexity for each PG algorithm and dataset. So, the reduction represents the percentage of instances that are included in the activity model of the original training set. For example, in the *ODI2* dataset contains 616 observations, if the training size is reduced by 95 %, the set of prototypes generated contains 31 samples, which will be included in the activity model of the NN approach. Furthermore, the accuracy average percentage and the reduction average percentage are indicated in Table 2.

The PG algorithms reduced the size of the training data and also reduced the computational complexity when classifying a new activity based on binary sensor-based activity recognition with a $1NN$ approach. The PG algorithms with a reduction around 95 %, significantly decreased the size of the initial training data. Therefore, the computational complexity is clearly reduced in the same proportion, i.e., 95 %, to classify a new activity.

Nevertheless, in some PG algorithms, the reduction of the size of the training data implies a reduction in the percentage accuracy. This fact is dramatic due to the fact that the accuracy average does not exceed 85 % in the following PG algorithms: *LVQ3*, *VQ*, *LVQTC*, *ENPC*, *GENN* and *MCA*. In these cases, the PG algorithms generalize the inconsistency, incoherence or noise in the dataset, implying negative results in the use of these PG algorithms to classify a new activity based on binary sensor-based activity recognition with a $1NN$ approach.

There are PG algorithms with an excellent performance in terms of accuracy percentage and reduction percentage of computational complexity in the four datasets. It is noteworthy that *MixtGauss* is the best PG algorithm that obtains an average percentage accuracy of 95 % with a reduction average percentage of 95 %.

Analyzing the results, we can point out that the PG algorithms: *MixtGauss*, *PSCSA* and *LVQPRU* obtain successful results in terms of accuracy percentage with the average accuracy above 90 %. This PG algorithms prefer numerical datasets, especially binary as the dataset used in this case study, which offers excellent reduction percentage without losing performance accuracy. Therefore, these PG algorithms used for activity recognition can be deemed as being very

Table 2. Results obtained of the PG algorithms with the four datasets

PG algorithms	Dataset	Accuracy (%)	Reduction (%)	Accuracy Av. (%)	Reduction Av. (%)
LVQ3	Odi1	49.92	95.00	72.00	95.00
LVQ3	ODI2	59.56	95.00		
LVQ3	VanKasteren	82.70	95.00		
VQ	Casas	88.33	95.00	68.63	95.00
VQ	ODI1	56.14	95.00		
VQ	ODI2	57.78	95.00		
VQ	VanKasteren	72.27	95.00		
LVQTC	Casas	94.17	95.00	76.89	95.00
LVQTC	ODI1	69.39	95.00		
LVQTC	ODI2	55.51	95.00		
LVQTC	VanKasteren	88.50	95.00		
MSE	Casas	93.33	63.00	86.50	86.75
MSE	ODI1	82.48	94.00		
MSE	ODI2	80.86	97.00		
MSE	VanKasteren	89.33	93.00		
HYB	Casas	93.33	95.00	81.95	95.00
HYB	ODI1	68.52	95.00		
HYB	ODI2	72.09	95.00		
HYB	VanKasteren	93.85	95.00		
LVQPRU	Casas	96.67	95.00	89.87	95.00
LVQPRU	ODI1	84.40	95.00		
LVQPRU	ODI2	84.58	95.00		
LVQPRU	VanKasteren	93.82	95.00		
PSCSA	Casas	86.67	95.00	91.98	96.25
PSCSA	ODI1	94.12	96.00		
PSCSA	ODI2	91.23	98.00		
PSCSA	VanKasteren	95.92	96.00		
ENPC	Casas	93.33	89.00	78.14	93.25
ENPC	ODI1	65.91	93.00		
ENPC	ODI2	68.02	96.00		
ENPC	VanKasteren	85.28	95.00		
GENN	Casas	95.00	95.00	73.89	95.00
GENN	ODI1	56.82	95.00		
GENN	ODI2	60.55	95.00		
GENN	VanKasteren	83.20	95.00		
MixtGauss	Casas	94.17	95.00	94.57	95.00
MixtGauss	ODI1	96.42	95.00		
MixtGauss	ODI2	93.84	95.00		
MixtGauss	VanKasteren	93.85	95.00		
MCA	Casas	6.67	95.00	67.36	96.00
MCA	ODI1	84.42	96.00		
MCA	ODI2	84.10	97.00		
MCA	VanKasteren	94.25	96.00		
POC	Casas	90.00	63.00	84.65	53.75
POC	ODI1	81.48	23.00		
POC	ODI2	72.87	70.00		
POC	VanKasteren	94.23	59.00		

useful given that a reduction in the number of stored instances corresponds to a reduction in the computational complexity, reducing the number of instances that are contained in the activity model.

4 Conclusions

This work has been focused on the identification of the prototype generation algorithms for the purpose of binary sensor-based activity recognition with NN approach in order to reduce the computational complexity of the classification process to be deployed in low cost devices. Twelve PG algorithms have been evaluated with four activity datasets. Results from the evaluation demonstrated the ability of the *MixtGauss*, *PSCSA* and *LVQPRU* PG algorithms to provide good performance and percentage reduction of approximately 95 % with an average accuracy percentage higher than 90 %.

Acknowledgments. This contribution has been supported by research projects: TIN2015-66524-P and UJAEN/2014/06/14. Invest Northern Ireland is acknowledged for partially supporting this project under the Competence Centre Program Grant RD0513853 - Connected Health Innovation Centre.

References

1. Chen, L., Hoey, J., Nugent, C., Cook, D.J., Yu, Z.: Sensor-based activity recognition. *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* **42**(6), 790–808 (2012)
2. Maksimovic, M., Vujovic, V., Davidovic, N., Milosevic, V., Raspberry, P.B.: Pi as internet of things hardware: performances and constraints. *Des. Issues* **3**(8), 8 (2014)
3. Doukas, C.: Building Internet of Things with the ARDUINO. CreateSpace Independent Publishing Platform, USA (2012)
4. Botta, A., Donato, W., Persico, V., Pescapé, A.: Integration of cloud computing and internet of things: a survey. *Future Gener. Comput. Syst.* **56**, 684–700 (2016)
5. Chui, M., Loffler, M., Roberts, R.: The Internet of Things (Process Optimization). McKinsey Quarterly, Germany (2010)
6. Gu, T., Wang, L., Wu, Z., Tao, X., Lu, J.: A pattern mining approach to sensor-based human activity recognition. *IEEE Trans. Knowl. Data Eng.* **23**(9), 1359–1372 (2011)
7. Li, C., Lin, M., Yang, L.T., Ding, C.: Integrating the enriched feature with machine learning algorithms for human movement and fall detection. *J. Supercomput.* **67**(3), 854–865 (2014)
8. San Martín, L.A., Pelaez, V.M., Gonzalez, R., Campos, A., Lobato, V.: Environmental user-preference learning for smart homes: an autonomous approach. *J. Ambient Intell. Smart Environ.* **2**(3), 327–342 (2010)
9. Chen, L., Nugent, C.: Ontology-based activity recognition in intelligent pervasive environments. *Int. J. Web Inf. Syst.* **5**(4), 410–430 (2009)
10. Chen, L., Nugent, C.D., Wang, H.: A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. Knowl. Data Eng.* **24**(6), 961–974 (2012)
11. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.* **13**(1), 21–27 (1967)

12. Wu, X., Kumar, V.: *The Top Ten Algorithms in Data Mining*, 1st edn. Chapman & Hall/CRC, Boca Raton (2009)
13. Al-Faiz, M.Z., Ali, A.A., Miry, A.H.: A k-nearest neighbor based algorithm for human arm movements recognition using EMG signals. In: *EPC-IQ 2010*, pp. 159–167 (2010)
14. Jafari, R., Li, W., Bajcsy, R., Glaser, S., Sastry, S.: Physical activity monitoring for assisted living at home. In: Leonhardt, S., Falck, T., Mähönen, P. (eds.) *BSN 2007. IFMBE*, vol. 13, pp. 213–219. Springer, Heidelberg (2007)
15. Moayeri Pour, G., Troped, P.J., Evans, J.J.: Environment feature extraction and classification for context aware physical activity monitoring. In: *SAS 2013*, pp. 123–128 (2013)
16. Kononenko, I., Machine, K.M., Learning, D.M.: *Introduction to Principles and Algorithms*. Horwood Publishing Limited, Chichester (2007)
17. Garcia, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype selection for nearest neighbor classification: taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(3), 417–435 (2012)
18. Lozano, M., Sotoca, J.M., Sanchez, J.S., Pla, F., Pekalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognit.* **39**(10), 1827–1838 (2006)
19. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: concepts and algorithms. *J. Mach. Learn. Res.* **10**, 747–776 (2009)
20. Garcia, E.K., Feldman, S., Gupta, M.R., Srivastava, S.: Completely lazy learning. *IEEE Trans. Knowl. Data Eng.* **22**(9), 1274–1285 (2010)
21. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Mach. Learn.* **38**(3), 257–286 (2000)
22. Geva, S., Site, J.: Adaptive nearest neighbor pattern classifier. *IEEE Trans. Neural Netw.* **2**(2), 318–322 (1991)
23. Kim, S.W., Oomenn, A.: A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Anal. Appl.* **6**, 232–244 (2003)
24. The, K.T.: The self-organizative map. *Proc. IEEE* **78**(9), 1464–1480 (1990)
25. Koplowitz, J., Brown, T.A.: On the relation of performance to editing in nearest neighbor rules. *Pattern Recognit.* **13**, 251–255 (1981)
26. Sanchez, J.S., Barandela, R., Marques, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. *Pattern Recognit. Lett.* **24**, 1015–1022 (2003)
27. Chang, C.L.: Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Comput.* **23**(11), 1179–1184 (1974)
28. Fayed, H.A., Hashem, S.R., Atiya, A.F.: Self-generating prototypes for pattern classification. *Pattern Recognit.* **40**(5), 1498–1509 (2007)
29. Raicharoen, T., Lursinsap, C.: A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm. *Pattern Recognit. Lett.* **26**(10), 1554–1567 (2005)
30. Sanchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognit.* **37**, 1561–1564 (2004)
31. Cook, D.J., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. *Method Inf. Med.* **48**(5), 480–485 (2009)
32. Synnott, J., Chen, L., Nugent, C.D., Moore, G.: The creation of simulated activity datasets using a graphical intelligent environment simulation tool, pp. 4143–4146 (2014)

33. Nugent, C., Synnott, J., Santanna, A., Espinilla, M., Cleland, I., Banos, L.J.O., et al.: An initiative for the creation of open datasets within the pervasive health-care, pp. 180–183 (2016)
34. Van Kasteren, T., Noulas, A., Englebienne, G., Krse, B.: Accurate activity recognition in a home setting, pp. 1–9 (2008)
35. Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L.: KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Multiple-Valued Logic Soft Comput.* **17**(2–3), 255–287 (2011)
36. Devijver, P.A., Kittler, J.: *Pattern Recognition: A Statistical Approach*. Prentice Hall, London (1982)