Application of a Cluster-Based Classifier Ensemble to Activity Recognition in Smart Homes

Anna Jurek, Yaxin Bi, Christopher Nugent, and Shengli Wu

School of Computing and Mathematics, University of Ulster,
Jordanstown, Shore Road, Newtownabbey, Co. Antrim, UK, BT37 0QB
jurek-a@email.ulster.ac.uk,
{y.bi,cd.nugent,s.wu1}@ulster.ac.uk

Abstract. An increasingly popular technique of monitoring activities within a smart environment involves the use of sensor technologies. With such an approach complex constructs of data are generated which subsequently require the use of activity recognition techniques to infer the underlying activity. The assignment of sensor data to one from a possible set of predefined activities can essentially be considered as a classification task. In this study, we propose the application of a cluster-based classifier ensemble method to the activity recognition problem, as an alternative to single classification models. Experimental evaluation has been conducted on publicly available sensor data collected over a period of 26 days from a single person apartment. Two types of sensor data representation have been considered, namely numeric and binary. The results show that the ensemble method performs with accuracies of 94.2% and 97.5% for numeric and binary data, respectively. These results outperformed a range of single classifiers.

Keywords: Activity recognition, classifier ensembles, smart homes.

1 Introduction

A popular approach in healthcare for assessing physical and cognitive well-being is through monitoring of users' activities of daily living (ADL). ADLs are activities which are performed daily, for example toileting, grooming, cooking or undertaking light house work. Monitoring such activities can provide useful information which can be used to either recognise an emergency situation or to identify behavioural changes over time. The problem of activity monitoring has been addressed by many studies over the years [1, 2, 3]. One of the key components of an activity recognition system is the use of sensor-based technology [2]. An environment can be equipped with sensors which have the ability to record a person's interaction within the environment itself, for example, recording whenever a cupboard is open or closed or the turning on or off of a domestic appliance. Based on the interactions captured it is possible to detect the change of state associated with an object/region within the environment. From a data analysis perspective it is therefore possible to infer from the change of a sensor's state that a person in the environment has interacted with a

C. Nugent, A. Coronato, and J. Bravo (Eds.): IWAAL 2013, LNCS 8277, pp. 88–95, 2013.

specific object. The output from such a sensorised environment is a stream of sensor activations that have occurred within a period of time. Analysis of the data can lead to the recognition of the activities being performed. From a computational perspective there are two main challenges to overcome. The first is related to the partitioning of the stream of data obtained from the sensors into segments which represent each of the activities [3]. Each activity is composed of a combination of actions, such as taking a cup from a cupboard and pouring water from a kettle. The second challenge relates to recognizing which of the predefined activities is represented by a given segmented stream of actions [4]. In other words, it can be regarded as a classification process of an instance representing a string of sensor activations into one of the classes representing activities such as cooking dinner or preparing a drink. It has been the focus of the current study to address the latter challenge with the aim of improving the overall accuracy of classification.

2 Relevant Work

A number of approaches to activity recognition, based on processing data obtained through low-level sensors, have been explored. They can be generally categorized as data-driven approaches and knowledge-driven approaches. In the former the most popular techniques adopted are classification models based on probabilistic reasoning for example Naïve Bayes [4], Hidden Markov Models (HMMs) [5], Conditional Random Fields (CRF) [11] and Partially Observable Markov Process (POMDP) [14]. Other algorithms such as Decision Trees [6] or Neural Networks [7] have also been considered. In the aforementioned studies these approaches have been reported as being successful, however, they require a large number of training examples. Within the application domain of smart environments there is, however, a lack of large annotated data sets. From a knowledge driven perspective the most popular approaches applied have been based on logical modelling [8] or evidential theory [1]. Knowledge-driven approaches do not require large data sets for training purposes, however, elicitation of the knowledge from the domain experts can be a challenging process.

As previously mentioned a large number of studies have been undertaken to improve the performance on the underlying approach to activity recognition. In this research it is hypothesized that ensemble methods could have an advantage over a single model applied to the problem of activity recognition. A classifier ensemble is a group of classifiers which are combined in some manner to produce, as an output, a consensus decision while classifying an unseen pattern [9]. The individual classifiers, which are combined to build the ensemble, are referred to as base classifiers. The main goal of building a classifier ensemble is to provide an improvement of classification performance in comparison to any single base classifier considered in isolation. Following the initial process of creating a collection of base classifiers the next step in the ensemble method is to combine the results obtained from each of the individual base classifiers. This combination process produces the final output and decision of the ensemble. Applying a number of different experts and averaging their decision decreases the risk of selecting the wrong classifier, from which a decision is to be made. Given that some activities may be represented by very similar sensor readings, for example preparing dinner or breakfast, it is beneficial to obtain a range of different opinions rather than applying a single model. In addition, some representation of activities may be very confusing given different human behaviours. For example, two activities may happen at the same time, or they can be interleaved. Representation of such an event may be classified as one of the two activities, depending on the subset of sensors (features) that are considered whilst making the decision. In most cases it is difficult to deal with such cases with a single classifier. Obtaining different opinions, for example, from classifiers trained with different subsets of features, may offer a better solution. In this work we propose a Cluster-Based Classifier Ensemble (CBCE) approach, which has already been presented as an effective classification technique [10], as an alternative approach for the purposes of activity recognition.

3 Cluster-Based Classifier Ensemble

With the CBCE approach a collection of clusters built on a training set is considered as one base classifier [10]. In the classification process a new instance is assigned to its closest cluster from each collection. The final decision is made based on the class labels of the instances from all the selected clusters. The CBCE approach has been previously evaluated on open data sets from the machine learning domain, however, it has not been previously applied within the field of activity recognition.

3.1 Creating Base Classifiers

To obtain a set of different base classifiers (collections of clusters), the clustering process is performed a number of times whilst varying two parameters. The first parameter varied is a subset of attributes applied while calculating the distance between 2 instances. The second parameter varied is the number of clusters generated in the clustering process.

The generation of a single base classifier can be presented as a 3 step process. In Step 1, the subset of features and the number of clusters that are going to be considered in the clustering process are selected randomly. In Step 2 all instances from the training set are divided into clusters according to the selected subset of features. As an output from this process a collection of clusters, which is considered as one base classifier, is obtained. For each cluster in the collection its centroid is calculated. It is assumed that each cluster supports one or more classes depending on the instances it contains. For example, if there is one instance assigned to class c in a cluster, we say that this cluster provides a degree of support to class c. The level of support allocated for a class is dependent on the number of instances from this class and the total number of instances that belong to the cluster. In Step 3 a matrix A_k, referred to as a support matrix, is constructed, where each row refers to one cluster and each column refers to one class. The values in the matrix represent the support given for each class by each of the clusters and are calculated as in Equation 1. N_{ij} represents the number of instances in cluster i that belong to class j and N_i represent the total number of instances in cluster i. M refers to the number of classes in the classification problem being considered. The entire process is repeated K times, where K refers to the size of the ensemble required.

$$A_k[i,j] = \begin{cases} \dfrac{N_{ij} - {N_i}/{M}}{N_i - {N_i}/{M}} & \textit{if} \quad N_{ij} - {N_i}/{M} \geq 0 \\ \\ \dfrac{N_{ij} - {N_i}/{M}}{{N_i}/{M}} & \textit{if} \quad \textit{otherwise} \end{cases} \tag{1}$$

3.2 Combining Base Classifier Outputs

The classification of a new instance can be presented as a 3 Step process. In Step 1, following the presentation of a new instance *x* the closest cluster from each collection, represented by one row of the matrix, is selected. The selection is performed based on the distance between the new instance and the centroid of the cluster. While calculating the Euclidean distance for each centroid only the subset of features applied in the clustering process is considered. Each of the selected clusters provides a level of support for each of the classes represented by values in the respective row from the support matrix. In Step 2, for each class c_j, the support provided by all selected clusters is combined through application of Equation 2:

$$ExSupp(c_j) = \sum_{k=1}^{K} \begin{cases} e^{\frac{A_k[i_k, j]}{1+d(x, x^k)}} & \textit{if} \quad A_k[i_k, j] > -1 \\ \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

where i_k is the row from matrix A_k representing the selected cluster, x_k refers to the centroid of the selected cluster and *d* represents the Euclidean distance metric. In Step 3 the class with the highest support is selected as the final decision.

One of the issues in sensor-based activity recognition is related with a situation where the same activity can be performed in many different ways, hence making it difficult to define a general description for each activity [13]. Classification models applied in activity recognition should therefore be able to deal with this situation. Given that the CBCE approach only considers the similarity between instances in the training and classification process, it is hypothesized that these approaches may have an advantage when dealing with this type of data. For instance-based classification methods there is no general definition required for each class (activity). A class label of a new instance is determined based on the class labels of some similar instances from the training set. Consequently, the most important concept is for representations of one class (activity) to be more similar with a specific representation than with the remaining alternatives. This can be satisfied, to a certain extent, by applying appropriate feature representations within the activity recognition problem.

4 Empirical Evaluation

For the purpose of this study a well known and publicly available data set[1] has been used. All information regarding the environment, sensors used and annotation applied during the data collection process can be found in [11]. Sensor data were collected over a period of 26 days in a 3- room apartment from a 26 year old male subject. Fourteen wireless sensors were installed in the apartment, each associated with one object: *'Microwave'*, *'Hall-Toilet door'*, *'Hall-Bathroom door'*, *'Cups cupboard'*, *'Fridge'*, *'Plates cupboard'*, *'Front door'*, *'Dishwasher'*, *'Toilet Flush'*, *'Freezer'*, *'Pans Cupboard'*, *'Washing machine'*, *'Groceries Cupboard'* and *'Hall-Bedroom door'*. Seven activities were observed throughout the duration of the experiments: *'Leave house'*, *'Use toilet'*, *'Take shower'*, *'Go to bed'*, *'Prepare breakfast'*, *'Prepare dinner'* and *'Get drink'*. In total there were 245 instances (activities) represented by 1,230 sensor events.

4.1 Data Pre-processing

In the activity recognition problem being considered instances are represented as a sequence of numbers/strings that may have different lengths. CBCE is an instance-based method that applies the Euclidean distance metric to calculate the distance between two instances. For this reason data to be used in the current study should be represented as vectors with the same dimension. The sensor recordings are initially converted into vectors of the same dimension. Consequently each instance (sequence of sensor labels) is represented by a 14-dimensional vector. Each dimension of the vector represents one sensor: [S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13]. In the experiments, numeric and binary representations of the sensor recordings are considered. For the numeric representation the position in the vector is an indicator of how many times the sensor appears in the sequence. For the binary representation, the value for each position is either 1 or 0 subsequently indicating if the sensor appears or does not appear in the sequence, respectively. As an example the activity [*Hall-Bathroom door*, *Toilet Flush*, *Toilet Flush*, *Toilet Flush*, *Hall-Bathroom Door*] in the numeric system will be presented as: [0,0,2,0,0,0,0,0,3,0,0,0,0,0]. We can read from this vector that sensors S3 (*Hall-Bathroom door*) and S9 (*Toilet Flush*) appeared in the sequence 2 and 3 times, respectively. The same activity in the binary system will be presented as: [0,0,1,0,0,0,0,0,1,0,0,0,0,0]. From the binary vector we can read that sensors S3 and S9 appeared in the sequence although we do not have any information relating to their number of occurrences.

4.2 Implementation Details

The clustering process with CBCE is performed by the *k*-means[2] algorithm implemented in Weka[3] that uses the Euclidean distance metric[4]. For each clustering process

[1] `http://sites.google.com/site/tim0306/`

[2] weka.clusters.SimpleKMeans.

the number of clusters to be generated was randomly selected with the lower bound equal to the number of classes in the classification problem being considered and the upper bound equal to three times the number of classes. Any empty clusters generated in the training process were automatically removed. The upper bound was enforced in an effort to decrease the chance of very small or empty clusters being generated. Its value was selected based on an evaluation of the clustering technique on training data. In future work we aim to consider the number of clusters as a function of 3 variables, namely size of the training set, number of classes and number of features applied in the clustering process. With CBCE the number of features is randomly chosen as a value between 1 and the total number of features. For each generated cluster its centroid[5] is identified. Each categorical/numerical feature of the centroid is calculated as the mode/average of the values of the features stemming from all instances within the cluster. For the size of the ensemble *K*=30 was selected, following the evaluation of the model on the training set. The CBCE approach was compared with 3 single classification algorithms implemented in Weka. The classifiers considered were Naive Bayes[6] (NB), J48 Tree[7] (J48) and *k* Nearest Neighbour[8] (*k*NN).

5 Results and Discussion

For the experiments, a 5-fold cross-validation was performed. The accuracy was calculated as an average percentage of the correctly classified instances out of all instances in the testing set. In addition to accuracy, all methods were evaluated using F-measure [14]. Two main issues were investigated. The first issue was related to the two types of activity representation, namely binary and numeric, that were applied in the experiments. The second issue was related to the evaluation of CBCE in the activity recognition problem. Results obtained by the 5 methods for numeric and binary data are presented in Fig. 1a and 1b, respectively. It can be observed from Fig. 1a and 1b that *k*NN and CBCE obtained better results when applied with binary, rather than numeric data representation. J48 performed at the same level with both types of data representation, while for the NB classifier the difference was marginal. For binary data *k*NN and CBCE obtained the highest accuracies, while for numeric data they were both outperformed by NB. For numeric data, *k*NN and CBCE obtained significantly lower values of F-Measure which is an indication that they did not perform equally well in each class. *k*NN and CBCE are based on a similar approach, where the classification decision is made based on the distances measured between a new instance and instances from the training set. We can infer from this that for the two methods, whilst calculating the similarity between the two activities, it is more important to know which actions have been performed rather than how many times each actions took place. This can be explained by the fact that in the classification problem being considered the same activity can be represented by different combinations of

[3] The Weka Data Mining Software: An Update SIGKDD Explorations, Volume 11, Issue 1.

[4] weka.core.EuclideanDistance.

[5] weka.clusters.SimpleKMeans.GetClustersCentroids.

[6] weka.classifiers.bayes.NaiveBayes.

[7] weka.classifiers.J48 –C 0.25 –M 2.

[8] weka.classifiers.lazy.IBk –K 1 –W 0 –X –A.

actions. For example, while cooking dinner the fridge may be opened a different number of times. This may cause some problems while calculating the Euclidean distance between the same activities that have been performed in 2 different ways. It may also appear that two instances from the same class will be considered as being very distant.

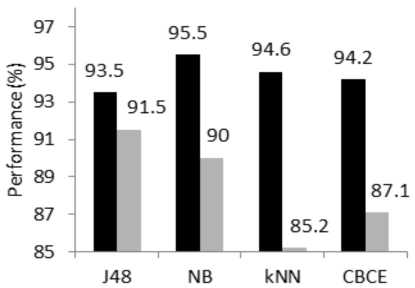

Fig. 1.a Percentage value of accuracy and F-Measure obtained for sensor data with numeric representation. J48 – J48 Tree, NB – Naïve Bayes, kNN–K Nearest Neighbour, CBCE-Cluster-Based Classifier Ensemble.

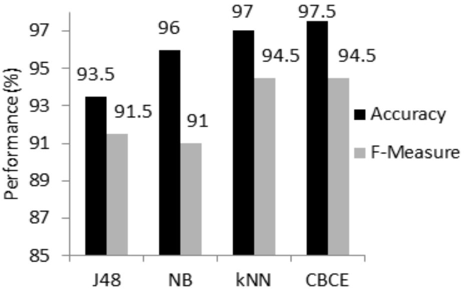

Fig. 1.b Percentage value of accuracy and F-Measure obtained for sensor data with binary representation. J48 – J48 Tree, NB – Naïve Bayes, kNN – K Nearest Neighbour, CBCE - Cluster-Based Classifier Ensemble.

Based on the results presented in Fig. 1.a and 1.b we can notice that the highest accuracy was obtained by CBCE (97.5%) and kNN (97%). Both of the methods outperformed NB (96%) and J48 (93.5%) in terms of accuracy and F-measure. This suggests that instance-based approaches are effective while applied in activity recognition problems. They not only obtained improved general accuracy, however, they also performed well in each class separately.

In addition to the accuracy it is, however, necessary to consider the computational cost of the methods. The training process of CBCE can be viewed as being complex as it requires clustering of the training set *K* different times. For large data sets this process may be time consuming. On the other hand, for a single *k*NN classifier no training is required. It should, however, be appreciated that following the training process, the classification of each new instance in the CBCE method is straight forward with limited computational cost. With the proposed approach a new instance needs to be compared with only a group of cluster centroids. For one base classifier the computational complexity can be estimated as $O(P \times l)$ where P and l represent number of clusters and features, respectively. For the *k*NN classifier, for number of instances in the training set equals N, the complexity can be estimated as $O(N \times l)$, which for large training data sets can be time consuming. We can therefore note that even though the CBCE approach requires a longer training process than *k*NN, it is, however, more efficient in terms of classification time. Once the ensemble is generated, classification of a new instance is a very simple task.

6 Conclusions

It can be concluded that instance-based methods are beneficial when applied in activity recognition problems compared to other classification techniques. Beside this, the experimental results demonstrate that instance-based classification methods perform better with binary rather than numeric representation of activities. This study provides a basis for further investigation into the application of ensemble methods in activity recognition within the application domain of smart homes. The new ensemble-based classification model was presented as being more accurate than a number of single classifiers. The study presented in the paper may be considered as early stage and further work is still intended. The first problem to be considered in the future work is an improved approach to selecting parameters for the model. It is presumed that appropriate selection of the number of clusters and the subset of features applied in the clustering process will improve the performance of the model.

References

1. Hong, X., Nugent, C.D., Mulvenna, M.D., McClean, S.I., Scotney, B.W., Devlin, S.: Evidential fusion of sensor data for activity recognition in smart homes. Pervasive Mobile Computing 5(3), 236–252 (2009)
2. Philipose, M., Fishkin, K., Perkowits, M., Patterson, D., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. IEEE Pervasive Computing Magazine 3(4), 50–57 (2004)
3. Rashidi, P., Cook, D., Holder, L., Schmitter-Edgecombe, M.: Discovering Activities to Recognize and Track in a Smart Environment. IEEE Trans. Knowl. Data Engineering 23(4), 527–539 (2011)
4. Tapia, E.M., Intille, S.S., Larson, K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
5. Hasan, M., Rubaiyeat, H., Lee, Y., Lee, S.: A HMM for Activity Recognition. In: 10th International Conference Advanced Communication Technology, pp. 843–846 (2008)
6. Logan, B., Healey, J., Philipose, M., Tapia, E.M., Intille, S.S.: A long-term evaluation of sensing modalities for activity recognition. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 483–500. Springer, Heidelberg (2007)
7. Yang, J.Y., Wang, J.S., Chen, Y.P.: Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. Pattern Recognition Letters, 2213–2220 (2008)
8. Chen, L., Nugent, C.D., Wang, H.: A Knowledge-Driven Approach to Activity Recognition in Smart Homes. IEEE Transaction on Knowledge and Data Engineering 24(6), 961–974 (2012)
9. Jurek, A., Bi, Y., Wu, S., Nugent, C.D.: A survey of commonly used ensemble-based classification techniques. Cambridge University Press (in press, 2013)
10. Jurek, A., Bi, Y., Wu, S., Nugent, C.D.: A Cluster-Based Classifier Ensemble as an Alternative to the Nearest Neighbour Ensemble. In: 24th IEEE International Conference on Tools with Artificial Intelligence, pp. 1100–1105 (2012)
11. van Kasteren, T.: Activity Recognition for Health Monitoring Elderly using Temporal Probabilistic Models. UvA Universiteit van Amsterdam, Ph.D. thesis (2011)
12. Powers, D.: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Machine Learning Technologies 2(1), 37–63 (2011)
13. Palmes, P., Pung, H.K., Gu, T., Xue, W., Chen, S.: Object relevance weight pattern mining for activity recognition and segmentation. Pervasive and Mobile Computing 6(1), 43–57 (2010)
14. Hoey, J., Plotz, T., Jackson, D., Monk, A., Pham, C., Olivier, P.: Rapid specification and automated generation of prompting systems to assist people with dementia. Pervasive and Mobile Computing 7(3), 299–318 (2011)