ClusterNN: A Hybrid Classification Approach to Mobile Activity Recognition

Sulaimon Bashir
Robert Gordon University
Aberdeen, UK
s.a.bashir@rgu.ac.uk

Daniel Doolan
Robert Gordon University
Aberdeen, UK
d.c.doolan@rgu.ac.uk

Andrei Petrovski
Robert Gordon University
Aberdeen, UK
a.petrovski@rgu.ac.uk

ABSTRACT

Mobile activity recognition from sensor data is based on supervised learning algorithms. Many algorithms have been proposed for this task. One of such algorithms is the K-nearest neighbour (KNN) algorithm. However, since KNN is an instance based algorithm its use in mobile activity recognition has been limited to offline evaluation on collected data. This is because for KNN to work well all the training instances must be kept in memory for similarity measurement with the test instance. This is however prohibitive for mobile environment. Therefore, we propose an unsupervised learning step that reduces the training set to a proportional size of the original dataset. The novel approach applies clustering to the dataset to obtain a set of micro clusters from which cluster characteristics are extracted for similarity measurement with new unseen data. These reduced representative sets can be used for classifying new instances using the nearest neighbour algorithm step on the mobile phone. Experimental evaluation of our proposed approach using real mobile activity recognition dataset shows improved result over the basic KNN algorithm.

Categories and Subject Descriptors

H.5.2 [**User/Machine Systems**]; I.5 [**Pattern Recognition**]: Metrics—*Percentage Accuracy*

Keywords

Activity Recognition, KNN, Smartphones, ClusterNN

1. INTRODUCTION

Mobile Activity Recognition has become a hot topic in recent years. This is due to its inherent usefulness in a wide range of applications. For example, the Goggle Android API now includes set of API that enables developers to use prebuilt models to recognise activity of users. Such activities being recognised include in-vehicle, on-bicycle, walking, still and running [4]. This can facilitate many types of context aware applications that enable a device to react based on user activity. For example, a device can be configured to increase the screen font size if the user is walking to make it easy to read the screen or a device may do self-management to switch to silent mode if user is driving. Similarly, activity recognition is useful for fitness and health monitoring [8], social networking [10] and commercial applications like activity based advertising [11]. Activity recognition is a classification task whereby labelled data are used to train a classification algorithm to induce a model that recognizes new unlabelled data. There are two approaches to model induction in mobile activity recognition [14]. The first approach called offline training collects sample data from subjects who perform the designated activities. The collected data is used to induce a model on a remote system off the mobile device. The induced model is later deployed into the application for recognition. The second approach called online training involves inducing the model directly on the device using the user's self-annotated data. The advantage of the second approach is that it can facilitate online and incremental learning for the model to adapt to changes in the environment. Many studies have evaluated different algorithms both in online and offline modes. Many of these studies have reported KNN to give good performance in terms of accuracy in offline training [3] [9][13]. But despite this performance, KNN is not being used for online recognition on mobile phone. The reason for this is due of the need to keep a large amount of data in memory for the instance based classification operation in KNN. This cost is prohibitive especially for the resource constraint and real time response requirement of the mobile device in the face of multitasking and multifarious mobile applications. Hence, there is need to make KNN amenable for online recognition of activities.

To make KNN amenable to online recognition with mobile activity recognition, we propose an offline data reduction step that reduces the amount of training instances to a desired percentage of the original dataset. The reduced set serves as a good representation of the training set and ensures better accuracy of KNN in an online setting for activity recognition. The evaluation of the proposed novel framework shows that it performs better than using the basic nearest neighbour algorithm.

The rest of this paper is organised as follows: Section II presents some of the related work in using KNN for activity recognition and other general algorithms. Section III describes the methodology. Section IV presents the result and discussion of the result. Section V gives the conclusion, and future work is highlighted in the last section of the paper.

MoMM 2015, December 11-13, 2015, Brussels, Belgium
© 2015 ACM. ISBN 978-1-4503-3493-8/15/12…$15.00
DOI: http://dx.doi.org/10.1145/2837126.2837140

2. RELATED WORK

Activity recognition using different sensor modalities and algorithms has been study extensively. A number of machine learning approaches in activity recognition were reviewed in [12]. A more recent review focusing on mobile phone based activity recognition is presented in [14]. The paper identifies many systems that are based on using smartphone sensors for activity recognition. Also, a comparative study of different classifier algorithms from Weka [5] machine learning tool was performed in [2] using data obtained from smartphone accelerometer. The data collected with phone placed in the shirt pocket was used to compare accuracies of IBK, Naive Bayes, Rotation Forest, VFI, DTNB and LMT algorithms while the data collected when the phone was placed in the palm position was used to compare accuracies of SMO, NNge, ClassificationViaRegression, FT, VFI, IBK and Naive Bayes algorithms. Out of all the algorithms tested, they reported IBK and IB1 to give the best accuracy for the hand's palm data and VFI gives the lowest accuracy. The KNN algorithm was not used directly on mobile phone for activity recognition. This can be attributed to the impracticability of using KNN directly for online activity recognition.

Similarly authors in [9][13] have all shown the superior performance of KNN in terms of accuracy for mobile AR in an offline evaluation scenario. Kose et al. [6] have proposed an improved KNN algorithm for online activity recognition. Initially, the training dataset consists of 4 features: average, minimum, maximum and standard deviation. The algorithm works by selecting k values from the minimum, maximum and average features across each activity data and the standard deviation of the data in each class. The reduced values and the corresponding class tags are employed during recognition phase. The main drawback of this approach is its feature dependence. The algorithm cannot be applied to a dataset with feature characteristics different from the one used in the algorithm. Our approach does not have this limitation as it is applicable to any feature set. Abdallah et. al. [1] have proposed a cluster based classification algorithm that clustered the training datasets into K clusters of the number of activities. The clusters obtained was refined by removing instances of other classes that were mixed-up in a given cluster having majority instances of another class. Then the cluster centroids were calculated. The algorithm employs four features computed from each cluster to classify new data. However, this algorithm does not treat new activity data to be classified as individual instances. Rather, it applies clustering to a window of raw accelerometer data and the clusters obtained are compared to the cluster generated from the training data using Euclidean distance, density, gravitational force and within cluster standard deviation. This approach does not segment between one activity and the other. In addition, the time required to collect enough samples that can meaningfully be clustered will be high for online recognition system that requires immediate and real time feedback of the recognised activity.

3. METHODOLOGY

Our proposed approach to make KNN amenable to online activity recognition employs a data reduction strategy to reduce the initial training set to a more compact set suitable for in-memory use for online recognition. As shown in Algorithm 1, the algorithm takes the training data and the desired percentage of data to retain as input and produces the Model Data (MD). The model data (MD) is the set of cluster centroid, minimum and maximum obtained after applying clustering on the dataset. In this algorithm, data samples belonging to each class$_i$ are clustered (lines 1-3 Algorithm 1) by applying a clustering technique on the data. Possible clustering algorithms include k-Means, DBScan, EM and host of others. However, for the sake of simplicity we employ Bisecting K-Means in the present work. After the clustering step, the list of cluster centres obtained for the class of data are stored in a list. In addition, we extracted the minimum and maximum data point from each cluster returned for the current class (Algorithm 1 lines 4-8). The number of clusters created per class is proportional to the number of samples in the class and the percentage of retention input to the algorithm. This step is repeated for each class in the data. Finally, the set of cluster characteristics i.e. centroids, minimum and maximum obtained from the different clusters of each class and their associated labels are returned from the algorithm. These represent the Model Data (MD) to be deployed for the online recognition on a mobile phone. The key feature of the model is that it is more compact and has a reduced resource overhead in terms of memory requirement and time when compared to the ordinary KNN. In addition, the reduced compact set including centroids can be adapted to evolving sensory stream as new unanticipated changes occurs in the input data distribution.

Algorithm 1: Offline Training

```
   Input: $\boldsymbol{C_n}$ number of classes in the dataset
   $\boldsymbol{K_n}$ percentage of data to remain in each classes of
   examples that serves as cluster centroids
   Data: $\boldsymbol{D = (x^i, y^i)}$ $\boldsymbol{x^i \in R^n}$ and $\boldsymbol{y^i \in R^1}$ set of
         training examples
   Result: MD=Centroid features
 1 foreach $class_k$ in $C_n$ do
 2     $data_{classi}$=getData(D, $class_k$)
 3     centroidsList[k], clusterAssign= Clustering(
       $data_{classi}$, $K_n$);
 4     foreach $cluster_k$ in $len(centroidsList_k)$ do
 5         $pointsInCluster_k$ = getData(clusterAssign,
           $cluster_k$)
 6         maximumList.append(max($pointsInCluster_k$))
 7         minimumList.append(min($pointsInCluster_k$))
 8     end
 9     MD = [centroidsList, minimumList,
       maximumList,$class_k$]
10 end
11 return MD
```

During the online phase, new instances can be classified by passing it and the MD to Nearest-Neighbour routine. It employs Euclidean distance to compute the K-nearest neighbour to the new instance and assigns the majority label of the K nearest point to it (Algorithm 2). Since we have more than one cluster characteristics in the MD, each is considered separately and a majority voting is performed on the outcome of each comparison. The final class given to the new instance is the majority label returned by all of them.

Algorithm 2: Online Classification

```
Input: $\boldsymbol{x_{new}}$ new unlabelled instance
$\boldsymbol{k}$ number of nearest neighbours
Data: $\boldsymbol{MD}$ compressed training set with
      characteristics features
Result: $y_{new}$=predicted class
foreach $clusterCharacteristics_i$ $in$ $MD$ do
    $prediction_i$
    =nearestNeighbour($clusterCharacteristics_i$, $x_{new}$,
    $k$ )
end
The output class is $(y_{new})$ =
$argmax_c$ $(prediction_c)$   $c = (1...C)$
return $y_{new}$
```

3.1 Experiments

In this section we describe the experiments conducted to evaluate the applicability and accuracy of the proposed algorithm. The dataset used in the experiment was the WISDM dataset released to the public for smartphone based activity recognition evaluations.

3.1.1 Dataset Description

The WISDM activity recognition dataset [7] was obtained from the accelerometer of mobile phones. The data were collected from 32 users that performed six designated activities of working, jogging, ascending and descending stairs, sitting and standing. Each data sample in the dataset is represented by 43 features. The features were obtained from the transformation of 200 raw samples of data recorded from the tri-axial accelerometer of a mobile phone. Each 200 worth of samples were recorded within a 10 second window with a sampling frequency of 20Hz. The features used were basic statistical features including standard deviation, average, resultants among others [7] described as follows:

- $X0..X9, Y0..Y9, Z0..Z9$ are set of bins of values representing fraction of accelerometer samples that fell within that bin.
- $XAVG, YAVG, ZAVG$ these features represent the average of the x, y, and z values in each recorded 200 samples.
- $XPEAK, YPEAK, ZPEAK$ these features approximate the dominant frequency along the x, y, and z axis values of the accelerometer within each 200 samples point.
- $XABSOLDEV, YABSOLDEV, ZABSOLDEV$ are the average absolute deviations from the mean value for each axis.
- $XSTANDDEV, YSTANDDEV, ZSTANDDEV$ are the standard deviations for each axis.
- $RESULTANT$ is the average of the square roots of the sum of the values of each axis squared $\sqrt{(x_i^2 + y_i^2 + z_i^2)}$

The dataset distribution spread across the six activities. The total samples in the obtained dataset and their distribution across each activity is shown in Table 1.

Table 1: Dataset Distribution

Activity label	**Instances**	**Percentage(%)**
Walking	2081	38.41
Jogging	1625	29.99
Upstairs	633	11.68
Downstairs	528	9.75
Sitting	306	5.65
Standing	245	4.52
Total	5418	100

3.1.2 Experimental Setup

We performed the experiment in two phases. In the first phase, we examined the accuracy of using individual cluster characteristic. We have three characteristics that were used for classification decision during the online phase (Algorithm 2). The centroid characteristic is the mean of the data points in a cluster. Minimum characteristic is the minimum values across each feature for the data points in a cluster while the maximum characteristic is the minimum values across each feature for the data points in a cluster. We varied the percentage of data retained ranging from 10-90% retention rate. We did not test the 100% retention rate because this will be equivalent to having all the dataset present. Therefore, we obtained the accuracies of using each characteristic as the number of data retention was varied. In the second phase of the experiment all the characteristics were combined to predict the classes of unseen instances used for testing the algorithms. The results obtained for each configuration of the experiment is presented in the next section.

In carrying out the experiment, we followed the hold-out evaluation strategy. The entire dataset was divided into the training set and test set. We ensured that the split of the data were proportionate in terms of the number of instances in each class for the training set and the testing set. The ratio of the split is 80 to 20%. The same configuration is used in evaluating ClusterNN and the benchmark algorithm

4. RESULTS AND DISCUSSION

The accuracy of the propose approach to classification of mobile activity recognition is presented here. Table 3, 5 and 6 show the results for the three different characteristics (centroid, minimum and maximum) employed individually by the nearest neighbour to classify new instances. As indicated in Table 3, centroid characteristic gives the overall best accuracy in classifying new instances when nearest neighbour is set to 1 and the percentage of data retained is 50%. The accuracy of using minimum characteristic for classification decision is the second best when K=1 and data retained is 80% while maximum characteristic gives the least accuracy when k=2 and data retained is either 80% or 90%. However, when we combined all the characteristics and used the majority voting scheme to select the final class of an instance after each characteristic has predicted a class, the best accuracy obtained is when K=2 and 80% data retention. These results indicate that there is a trade-off between accuracy and the amount of data retained for classification across each of the three characteristics and the combined characteristic. Thus, we can select the percentage of data reduction based on the level of desired accuracy. We therefore, adopt 50% as the optimal data retention given that the overall best accuracy of 81.46% is achieved with centroid characteristic at this point. More so, going beyond this percentage of data reduction did not yield any high significant increase in accuracy across each of the characteristic and their combination. Therefore, going beyond 50% data retention is ineffectual considering that the corresponding accuracy is insignificant.

Table 2 shows the comparison of the accuracy of KNN which utilized all the dataset and the various characteristics with only 50% data retention. We can see that the accuracy of basic KNN is lower than the centroid characteristic at this point. Since other characteristics utilized lower dataset, their accuracy can be traded-off for the smaller amount of data required when compared to the basic KNN.

Using this small sample and nearest neighbour set to 1 will have a minimal resource overhead during online recognition compared to using the basic KNN algorithm with all the training data. Figure 1 shows the relative accuracy of each characteristic with varying number of nearest neighbours for the 50% data retention. As the figure shows, the accuracy for each measure decreases with increasing number of K. However, the best accuracy for KNN is obtained when K is 2 and decreases afterwards as well. The general low accuracy below 90% in all the experiments can be attributed to the nature of the dataset. The dataset contain data from 32 different users of varying characteristics in performing the designated activity. This produces many variations in the training and testing data. Nevertheless, the performance of our centroid characteristic is good given the fact that it can use a reduced dataset for online recognition compare to KNN that requires the entire training instance to achieve good performance.

Table 2: Accuracy of the ClusterNN Algorithm with Different Cluster Characteristics Compared with KNN Algorithm

K	Combined	Centroid	Maximum	Minimum	KNN
1	**79.80**	**81.46**	77.31	**77.95**	80.90
2	78.23	79.70	**77.40**	77.58	**80.99**
3	77.58	79.43	76.11	77.68	79.89
4	77.68	79.24	77.03	77.86	80.07
5	77.58	78.51	76.38	77.49	79.98

Table 3: Accuracy of Using Centroid Characteristics with Varying Percentage of Data Retained

					Centroid				
K	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
1	**76.85**	**79.80**	**80.07**	**80.54**	**81.46**	**80.81**	**80.90**	**80.63**	81.00
2	76.01	77.77	78.78	79.80	79.70	80.26	80.63	**81.09**	80.81
3	75.83	78.32	78.78	78.32	79.43	79.24	80.17	79.70	78.60
4	76.20	77.31	77.95	78.60	79.24	78.14	79.34	79.61	79.61
5	75.37	76.57	77.58	78.04	78.51	78.14	79.34	79.34	79.43

Table 4: Accuracy of Using Combined Characteristics with Varying Percentage of Data Retained

					Combined				
K	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
1	**73.71**	**76.85**	**78.32**	**79.61**	**79.80**	**81.18**	**80.81**	81.09	80.35
2	69.74	72.79	76.01	78.14	78.23	79.34	80.07	**81.18**	**81.00**
3	70.20	73.80	75.74	76.66	77.58	78.14	79.52	80.26	79.24
4	70.39	73.43	75.55	76.66	77.68	77.58	79.98	80.26	79.89
5	69.46	72.88	75.18	76.38	77.58	77.49	79.06	79.06	79.80

Table 5: Accuracy of Using Minimum Characteristics with Varying Percentage of Data Retained

					Minimum				
K	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
1	70.48	71.59	75.09	78.04	**77.95**	**79.89**	**79.98**	**81.73**	80.44
2	69.10	71.49	75.37	77.03	77.58	79.15	79.52	80.72	**80.90**
3	69.37	73.43	75.83	76.57	77.68	78.97	78.78	79.80	79.61
4	69.83	73.62	75.55	77.31	77.86	78.23	79.61	80.72	80.35
5	69.46	72.42	75.00	76.94	77.49	78.14	78.41	79.52	79.98

Table 6: Accuracy of Using Maximum Characteristics with Varying Percentage of Data Retained

					Maximum				
K	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
1	62.55	65.31	73.15	75.27675	77.31	78.87	79.15	79.34	79.98
2	62.92	65.22	72.05	75	**77.40**	**79.06**	**80.07**	**80.44**	**80.44**
3	64.11	66.42	72.05	73.80074	76.11	77.40	77.77	79.34	78.78
4	63.10	65.87	71.77	74.53875	77.03	77.95	79.52	78.60	79.43
5	63.10	66.61	71.31	73.70849	76.38	76.75	79.06	78.51	79.43

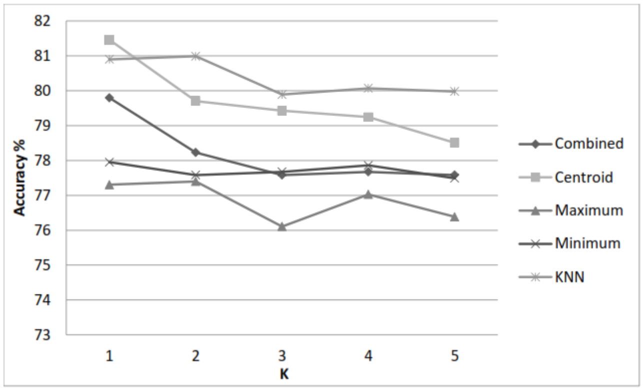

Figure 1: Accuracy of Using Different Cluster Characteristics and KNN Algorithm

5. CONCLUSION

In this paper we identified the drawback of using KNN for online activity recognition on mobile phone. The drawback of KNN in terms of keeping all the large amount of training data available at recognition time is addressed by proposing a hybrid approach to classification. The algorithm is based on the concept of clustering and nearest neighbour. The novel approach employs bisecting k-Means algorithm to cluster the training instances into k clusters per class. The number of clusters per class is computed proportionally to the number of samples in each class to ensure a balance proportionate of the instances in each cluster. The evaluation of the approach shows that it performed better than basic KNN on a realistic mobile activity recognition dataset.

6. FUTURE WORK

The results show that our approach performs better than KNN. Since we performed the test using hold-out approach we are able to show the true accuracy of the algorithm on totally unseen data. In the future, we will improve the performance of the algorithm by employing other clustering techniques. In addition, online stream clustering method will be considered to see the possibility of performing the preprocessing step of data reduction online. In addition, further investigation into the resource usage of our algorithm on the mobile phone will be conducted and compare with KNN to show the benefits of reduced dataset on resource consumption.

7. ACKNOWLEDGMENTS

This research was supported by the National Information Development Fund (NITDEF), Nigeria under the office of the National Information Development Agency, Abuja, Nigeria.

8. REFERENCES

[1] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. Cbars: Cluster based classification for activity recognition systems. In *Advanced Machine Learning Technologies and Applications*, pages 82–91. Springer, 2012.

[2] M. A. Ayu, S. A. Ismail, A. F. A. Matin, and T. Mantoro. A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognition. *Procedia Engineering*, 41:224–229, 2012.

[3] S. A. Bashir, D. C. Doolan, and A. Petrovski. The impact of feature vector length on activity recognition accuracy on mobile phone. *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2015, 1-3 July, 2015, London, U.K.*, 1:332–337, 2015.

[4] Google. https://developers.google.com/.../activityrecognitionapi, Accessed: 10th August 2015.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[6] M. Kose, O. D. Incel, and C. Ersoy. Online human activity recognition on smart phones. In *Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, pages 11–15, 2012.

[7] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[8] N. D. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell. Bewell: A smartphone application to monitor, model and promote wellbeing. In *5th International ICST Conference on Pervasive Computing Technologies for Healthcare*, pages 23–26, 2011.

[9] S. L. Lau and K. David. Movement recognition using the accelerometer in smartphones. In *Future Network and Mobile Summit, 2010*, pages 1–9, June 2010.

[10] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM, 2008.

[11] K. Partridge and B. Begole. Activity-based advertising techniques and challenges. In *Proceedings of Workshop on Pervasive Advertising*, 2009.

[12] S. J. Preece, J. Y. Goulermas, L. P. Kenney, and D. Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*, 56(3):871–879, 2009.

[13] Z. Prekopcsák, S. Soha, T. Henk, and C. Gáspár-Papanek. *Activity recognition for personal time management*. Springer, 2009.

[14] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.