Reducing the intrusion of user-trained activity recognition systems

William Duffy, Kevin Curran, Daniel Kelly, Tom Lunney
School of Computing, Engineering & Intelligent Systems
Ulster University
Derry, United Kingdom
Duffy-w@ulster.ac.uk

***Abstract*—Many supervised activity recognition systems require a fully labelled time-series for accurate classification. However, gathering these labels is a difficult and often unrealistic task, especially over long-time frames or outside of laboratory conditions. A potential solution is through diary studies, allowing for a user-trained activity recognition system. Requests will be presented on the user's smart device and while this approach will be significantly less intrusive than current methods, frequent or inappropriately timed requests could reduce user acceptance. This paper proposes to further reduce user intrusion by making a prediction about the next user request and analyzing the classifiers confidence in this prediction. Two methods are presented, and with careful selection of the confidence threshold, they resulted in up to a 35% reduction in user requests with a minimal reduction in accuracy.**

***Keywords*—activity recognition, Random Forest, ECOC-SVM, experience sampling**

I. Introduction

As an important part of ubiquitous computing, activity recognition is an active research area. Uses are found in multiple areas including healthcare, where activity recognition can be used for monitoring [1] and rehabilitation [2] of patients and sports by tracking exercise [3].

The collection of data annotations is a prominent issue in activity recognition systems [4]. Many sensing devices have high refresh rates and collecting sufficient ground truth for reasonably accurate classification of multiple activities is a considerable task. Attempts to solve this problem include the combination of diary studies with weakly supervised machine learning algorithms [4]. However, requesting excessive input from a user could become intrusive.

This work focuses on increasing the usability of activity recognition systems which require user trained data capture requests. Usability can be increased by reducing the frequency of requests or increasing the time between requests. Two methods are proposed, both estimating the confidence in a prediction by analyzing an ensemble learners votes. When the confidence is above a threshold, the learner will use its prediction but if the confidence is below a threshold then the system will simulate a user request. One potential advantage this brings is that with repeated use it is likely that the system will become more confident and therefore capable of making accurate predictions. However, a potential issue could be the computational intensity required for online classification of a constant stream of data, and the subsequent retraining of classifiers with these newly labelled data points.

The methodology of the experiments will be presented in section III, results will be presented in section IV and the conclusion presented in section V.

II. Related Work

As a diary study, experience sampling has been used to quantify the mood and feelings of an individual. Primarily used in the medical field, it works by requesting data from users at intervals. Currently, performing activity monitoring through questionnaires where patients have to recall activities results in lower accuracy rates relative to a laboratory study [5]. As experience sampling is an in the moment request, it does not require individuals to remember anything but just to record their actions or feelings at the moment of request [6].

Previous work in activity recognition has focused on fixed length experience sampling request windows [4]. While multiple window lengths ranging from 10 to 180 minutes were tested, the system will always make a request once the experience sampling time has expired. Weak supervision techniques such as Multiple Instance Learning [4], Graph Label Propagation [4] and Active Learning [7] are then used to populate labels to the unlabeled data points in between the request windows. While this is less intrusive than typical activity recognition systems which require each data point to be labelled, it may become unacceptable to a user over time.

Methods have also been presented which populate labels to unlabeled data points based on their distance in a feature space. These methods in combination with experience sampling managed to reduce label requirements by 99.8% while maintaining acceptable levels of classification accuracy, but do not reduce the intrusion of repeated data requests [8].

III. Methodology

A. Dataset

The human Activity and Postural Transitions (HAPT) dataset [9] allows us to move away from laboratory based studies of previous activity recognition experiments and focus on real world usability as most people now carry some form of smart device. It uses embedded smartphone sensors for signal

978-1-5386-6046-1/18/$31.00 ©2018 Crown

capture, namely an accelerometer and gyroscope sampling at 50Hz.

The human activities performed are standing, lying, sitting, walking and walking upstairs/downstairs. The postural transitions between the static activities are recorded but are not used as they are not a fit for this experiment.

With 30 participants ranging in age from 19-48 years, validation is performed with a user-independent 70%/30% train/test split. As it is user independent the system will not be trained on the same individuals that it is tested on.

B. Feature Extraction

The feature set used is pre-computed. The raw signals from the embedded sensors are converted into several different time series including "body", "gravity" and signal Jerk. A 561-width feature vector is then calculated from this time-series, each containing 2.56 seconds of activity data. Example features include signal energy, interquartile range, and skewness.

C. Experience Sampling

The experience sampling will simulate the collection of input from a user. In a real world scenario, a prompt for information would be presented on the user's device, however, as this experiment is being performed on previously captured data the requests are simulated. This is accomplished by accessing the full ground truth and obtaining a label for the current feature vector.

The sampling window is going to be made up of 1, 5, 10, 15 and 20-minute intervals. While a longer sampling window is going to be better from a real-world feasibility perspective, it is unlikely that enough data is available in the dataset.

Before the system can begin to estimate its confidence in future data points it will first need to gather some initial data. This is performed by having a "training phase". This training phase will simulate the collection of 5 annotations at 5-minute intervals. The same training phase will be applied to each size of experience sampling window, so each will have the same starting data. It has been shown in previous studies that many users are fine with completing a short training segment if it reduces future interruptions [10].

An obvious problem with this method of data collection is that the device moves when the user is going to input data. For example, if the user is currently sitting and the device requests input. The motion of moving the phone from a pocket and typing the name of the activity would be incorrectly labelled as sitting. Potential solutions to this problem are to detect an extended period of the same activity and make a request which states "what activity were you performing X minutes ago?"

D. Confidence

Two methods are used for confidence classification. One which uses reduces the number of requests and another which attempts to classify each individual data point while maintaining a minimum distance between data requests.

For each of these methods, the Random Forest classifier [11] is first trained on the labelled feature vectors gathered from the training phase. To ensure that each experiment is repeatable, the random seed has been set to 1. Classifier confidence can then be inferred by averaging these votes, and the classifier retrained each time a new label is gained. While likely to result in some noisy labels, tuning of a threshold value will be key. The ideal value being low enough to allow the system to make predictions but also high enough to reduce the likelihood of incorrect labels.

The systems earliest predictions are those most likely to be incorrect. This is because the system will not yet have much data. To reduce this problem each method will also use a weighted threshold. Increasing the required threshold during early predictions and as the system learns more, the threshold will be reduced.

1) Method one

This method attempts to reduce the number of user requests by making a prediction at each of the experience sampling request points. The request points will be at fixed times, for example, every 5 minutes. At each of these requests, the classifier will make a prediction and if the average of the votes of this prediction are above a threshold value then it will be accepted. Otherwise, it will request input from the user.

2) Method two

This method, instead of looking at data points in fixed sampling windows, looks at each individual data point. Confidence will be measured of each data point and if confidence is below the threshold user input will be requested. Limitations will be put on the number of requests made, and the system will only be able to make a request if a minimum period of time has passed since the previous request.

It is likely that a higher confidence threshold will be required for this method as significantly more data points are being checked, bringing a higher chance of noisy labels.

E. Classification

Previous work into activity recognition has had success using Support Vector Machines (SVM) for classification. As this work uses several activities, it would be advantageous to have a multi-class SVM. For this reason, an error-correcting output codes SVM (ECOC SVM) is being used.

IV. Experiments

A. Full Experience Sampling

To find the maximum possible accuracy experience sampling alone can achieve, the experiment will be run without either method active. A sampling window of 0 represents standard supervision with each feature vector having a label.

Method One – 5 Minute Sampling Window			
Confidence	**User Requests**	**Predictions**	**Incorrect Predictions**
50%	**23**	**36**	**18**
60%	**43**	**16**	**2**
70%	**49**	**10**	**0**
80%	**56**	**3**	**0**
90%	**59**	**0**	**0**

Method Two - 5 Minute Sampling Window			
Confidence	**User Requests**	**Predictions**	**Incorrect Predictions**
50%	**7**	**6719**	**5617**
60%	**30**	**6472**	**4104**
70%	**53**	**3638**	**1209**
80%	**56**	**1643**	**364**
90%	**58**	**120**	**0**

Table 1 – Tuning the prediction threshold

Sampling Window Length	0	1	5	10	15	20
Number of labels	**7415**	**302**	**64**	**34**	**24**	**19**
Accuracy	96.5%	91.1%	84.4%	73.5%	73.8%	71.1%

Table 2 – Full experience sampling results

As is visible from table 2, the shorter experience sampling windows performed better as more data is available. A 1-minute sampling window significantly reduces the number of labels required while maintaining an accuracy of 91.1%

Finding the ideal value for the prediction threshold will be key for ensuring the best results. It will be tuned individually for each method, as the second method is likely to require a much stricter value. For both methods, the threshold will be tuned to the 5-minute sampling window, as it provides significantly less data than standard supervised and is also unlikely to run into insufficient dataset problems.

a) Method one

The results in table 1 show that having a very high threshold will reduce the number of incorrect predictions, but it also reduces the overall number of predictions made. While 60% has reduced the number of requests by the largest amount, it has resulted in two incorrect predictions. 70%, however, has still reduced the number of predictions by 17% and has not made any incorrect predictions.

b) Method two

As is visible in table 1, the confidences required for the second method are higher. Anything below 90% results in an unacceptably high number of incorrect predictions which are likely to cause classification problems.

While 90% only reduces the overall number of user requests by 1 relative to method one at this threshold, it provides an extra 120 successfully labelled feature vectors which could potentially increase classification accuracy.

B. Results

The following section will provide a comparison of the results of the two methods running on the HAPT dataset.

1) Method one

Table 3 shows that with the lower sampling window lengths, the number of predictions is successfully reduced. The 1, 5 and 10-minute sampling windows having the number of requests reduced by 35%, 16%, and 9% respectively.

The longer windows showed less success, with the 15-minute window only providing 1 correct prediction and the 20-minute window providing no correct predictions. It is possible however that these problems are being caused by the length of the dataset.

When compared with the results found in table 2 we can see that the results of this method are almost identical to the full experience sampling classification accuracies. The 2 incorrect predictions on the 1-minute sampling window only resulted in an overall accuracy reduction of 0.1%.

2) Method two

Table 3 shows that the second method provides more data for the classifier when the sampling window is kept to 10-minutes or below. In the 1, 5 and 10-minute sampling windows it provides 1546, 120 and 459 extra features for the classifier. Like method one, the longer 15+ windows gained no benefit from this method. Similar to the first method, the number of user requests relative to experience sampling was reduced but not by the same margin.

3) Comparison

Both the first and second method managed to reduce the number of user requests relative to experience sampling. However, method two has provided the benefit of significantly more labelled feature vectors providing a slight increase in accuracy.

With the 5-minute window length, method two has reduced the number of requests by 9% and has gathered a much higher number of successful predictions than method one. Unusually though the classifier accuracy has gone down. This could be because while the experience sampling and method one collect the same labelled feature vectors, the second method can pick different feature vectors for labelling. It is possible that the feature vectors selected provided less discriminative data.

The 10-minute window lengths have a similar number of

Method One					
Sampling window length	**1**	**5**	**10**	**15**	**20**
Total sampling requests	302	64	34	24	19
User requests	191	49	26	18	11
Predictions	106	10	3	1	3
Incorrect predictions	2	0	0	0	3
Classifier Accuracy	91.0%	84.4%	73.5%	73.8%	66.7%

Method Two					
Sampling window length	**1**	**5**	**10**	**15**	**20**
Total sampling requests	1844	183	491	21	16
User requests	286	58	27	16	11
Predictions	1553	120	459	0	0
Incorrect predictions	7	0	0	0	0
Classifier Accuracy	91.2%	79.6%	76.8%	72.3%	63.2%

Table 3 – Prediction and Classification results

1 Minute Sampling Window						
Activity	**Walking**	**Walking Upstairs**	**Walking Downstairs**	**Sitting**	**Standing**	**Laying**
Full Experience Sampling	99.6%	90.2%	85.2%	80.5%	89.6%	99.8%
Method One	97.8%	91.3%	85.7%	80.5%	89.6%	99.8%
Method Two	98.8%	95.1%	87.6%	68.5%	96.2%	99.8%

10 Minute Sampling Window						
Activity	**Walking**	**Walking Upstairs**	**Walking Downstairs**	**Sitting**	**Standing**	**Laying**
Full Experience Sampling	94%	67.1%	71.9%	98.4%	14.2%	98.7%
Method One	94%	67.1%	71.9%	98.4%	14.2%	98.7%
Method Two	88.9%	91.1%	33.8%	78.9%	61.7%	99.8%

Table 4 – Individual activity results

requests for both methods, however, method two made 459 more successful predictions than method one. These extra predictions also increased the classifier accuracy slightly.

Longer windows (15+ minutes) simply do not have enough information for sufficient training of the classifier. The first method managed to make a few predictions, however, most of them are wrong. The second method managed no predictions for the longer windows, likely caused by the combination of the high threshold plus the lack of training data meaning the classifier will never be confident in any predictions.

The first method performs very well in reducing the number of user interruptions, especially at the lowest experience sampling window lengths. At the one minute window length it has reduced the number of requests by 35% with minimal effect on accuracy. However, the second method performs better at classification and prediction. It reduced the number of requests relative to experience sampling but not by the amount that method one managed. It is possible in a real world scenario that the number of requests made will decrease quicker with the second method due to the amount of extra data it is providing to the classifier. Another issue noted is the performance cost of the second method, as it is continually providing an online prediction of each data point and retraining the classifier where required. Its performance impact is significant. This could be a serious problem on smart devices as it will impact both the performance of the device and its battery life.

Table 4 shows the classification accuracy for each individual activity for the 1 and 10-minute experience sampling windows. For the 1-minute windows, we can see that method one matches or exceeds the classification accuracy of the experience sampling method and this is achieved with significantly less user intrusion. The second method managed to exceed the performance of method one and experience sampling in all but sitting and laying activities. Although not reducing user intrusion as much as method one, it still uses 5% fewer requests than full experience sampling. For the 10-minute sampling window, method one and experience sampling performed the same, except method one achieved this performance with 9% fewer user requests. The second methods results are difficult to compare as it would have found different feature vectors than the fixed feature vectors of the experience sampling and method one. It performed better on walking upstairs, standing and laying, but the remaining activities have seen reductions, especially walking downstairs.

V. Conclusion

This paper presents a solution to the problem of user interruptions in diary study methods of activity recognition. Two methods have been developed which attempt to solve the problem in separate ways. The first method has shown an up to 35% decrease in the number of user requests while maintaining comparable levels of classification accuracy to methods which have higher levels of user intrusion. The second method also reduced the number of requests but not to the degree of the first method. It did, however, provide significantly more labelled feature vectors which increased the classification accuracy in some instances.

As the intended usage of these methods is on smart devices where users can be provided with a minimally intrusive personalized system of activity recognition, the second method will need to be refined to reduce its performance impact. Potential ways of doing this could be through reducing the number of feature vectors which are predicted upon or only retraining the classifier after a certain number of predictions have been made. Future work could look at methods which can predict the state of the user to further reduce intrusion. These could be capturing data from the user about their tolerance to interruption and attempting to find activities or time frames where it is appropriate to ask for input.

VI. References

[1] S. Jiang *et al.*, “CareNet: An Integrated Wireless Sensor Networking Environment for Remote Healthcare,” in *Proceedings of the 3rd International ICST Conference on Body Area Networks*, 2008.

[2] E. J. W. Van Someren *et al.*, “A new actigraph for long-term registration of the duration and intensity of tremor and movement,” *IEEE Trans. Biomed. Eng.*, vol. 45, no. 3, pp. 386–395, 1998.

[3] N. Alshurafa *et al.*, “Designing a Robust Activity Recognition Framework for Health and Exergaming Using Wearable Sensors,” *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 5, pp. 1636–1646, 2014.

[4] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, “Weakly supervised recognition of daily life activities with wearable sensors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2521–2537, 2011.

[5] R. J. Shephard, “Limits to the measurement of habitual physical activity by questionnaires,” *Br J Sport. Med*, vol. 37, pp. 197–206, 2003.

[6] S. Consolvo and M. Walker, “Using the experience sampling method to evaluate ubicomp applications,” *IEEE Pervasive Comput.*, vol. 2, no. 2, pp. 24–31, 2003.

[7] M. Stikic, K. Van Laerhoven, and B. Schiele, “Exploring semi-supervised and active learning for activity recognition,” *2008 12th IEEE Int. Symp. Wearable Comput.*, pp. 81–88, 2008.

[8] W. Duffy, K. Curran, D. Kelly and T. Lunney “Addressing the problem of Activity Recognition with Experience Sampling and Weak Learning,” in *In IntelliSys 2018 - Proceedings of 2018 SAI Intelligent Systems Conference.*, 2018. (Accepted, awaiting publication)

[9] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-Aware Human Activity Recognition Using Smartphones,” *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[10] A. Kapoor and E. Horvitz, “Experience sampling for building predictive user models,” *Proceeding twenty-sixth Annu. CHI Conf. Hum. factors Comput. Syst. - CHI ’08*, pp. 657–666, 2008.

[11] Leo, “Random Forest,” 2012. [Online]. Available: https://uk.mathworks.com/matlabcentral/fileexchange/31036-random-forest.