A Mobile Application to Detect Abnormal Patterns of Activity

Omar Abdul Baki, Joy Zhang, Martin Griss, and Tony Lin

Carnegie Mellon Silicon Valley,
NASA Research Park, Bldg. 23, Moffet Field, CA 94035
{omar.abdulbaki,joy.zhang,martin.griss,tony.lin}@sv.cmu.edu

Abstract. In this paper we introduce an unsupervised online clustering algorithm to detect abnormal activities using mobile devices. This algorithm constantly monitors a user's daily routine and builds his/her personal behavior model through online clustering. When the system observes activities that do not belong to any known normal activities, it immediately generates alert signals so that incidents can be handled in time. In the proposed algorithm, activities are characterized by users' postures, movements, and their indoor location. Experimental results show that the behavior models are indeed user-specific. Our current system achieves 90% precision and 40% recall for anomalous activity detection.

Keywords: activity monitoring; context-aware; abnormality detection; unsupervised learning; online clustering; senior care.

1 Introduction

As the baby boomer generation ages, the US census predicts a dramatic increase in persons older than 65 from 44 million in 2005 to 74 million by 2020 [1]. To cope with the massive care needs of an aging population, there must be a shift in the paradigm of care in the US. On the one hand, as people age, they would like to maintain a certain degree of independence to maintain their quality of life [2]. On the other hand, in order to allow for quality care for each person, care professionals must be able to keep track of multiple patients' daily life routines and abnormalities. Abnormal activities such as falling often result in severe medical situations. In fact, accidental falls caused more deaths than all other medical situations among seniors [3].

Existing activity monitoring systems are costly and invasive. Some solutions use wearable devices such as bracelets, pendants, or devices mounted on the waist to detect falls and monitor vital signs. Other solutions use sensors such as cameras, motion sensors, or infrared cameras to monitor vital signs, detect falls, and even recognize certain high level activities.

Most existing monitoring systems are designed to detect only certain classes of events. For example a device which detects falls may not be able to detect if an individual is still in bed at an unusual time. As a result these other types of accidents or problems may end up undetected.

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 190–202, 2010.

In this paper, we describe the Mobile Activity Monitoring System (MAMS), a novel approach to home care monitoring based on detecting anomalous activities using mobile devices. We assume that accidents and other critical events are "abnormal" compared to a user's "normal" activities. We learn each user's normal patterns of activities and flag anomalous behaviors which are significantly different from a user's daily routine. Using a mobile device, we build a user's behavior model. Behavior is characterized by a user's posture, movement, and location. For example, an activity can be represented as "climbing" on a "stairway". In this paper, we use the built-in tri-axial accelerometer in the Nokia N95 mobile phone to infer user posture and movement, and Wi-Fi based positioning to estimate a user's indoor location. These elements are used as features for the clustering K-nearest neighbor classifier.

2 Unsupervised Anomaly Detection

Most activity recognition systems are supervised. They are trained using a set of labeled activity examples. Supervised methods have three problems in activity monitoring. Firstly, creating the labeled training data is time-consuming. This is especially the case for applications such as monitoring the elderly. It is impossible to enumerate all activities and collect data for each activity, especially for abnormal activities such as "falling down a flight of stairs."

Secondly, supervised learning solutions are limited to detecting activities defined in the training data set. Activities that are not defined in the training data will not be detected.

Thirdly, activities are user-dependent and context-dependent. People have different life styles and different daily routines; thus an activity considered to be normal for one individual may not necessarily be so for another person. Activities also depend on context. "Falling down onto a bed" could be normal but "falling down in the bathroom" should flag an alert message. Context information such as location, movement and posture are user-dependent information and can not be applied directly to other users. Thus, user-independent behavior models obtained from supervised training might not generalize well for all users.

K-nearest neighbor clustering provides for scalable activity monitoring since no labeling is required for any of the data points collected. This allows MAMS to collect data at a higher rate and larger volume than any supervised algorithm. It also achieves our desired goal of creating a personalized system capable of learning user-specific activities. More specifically users' behavior models will be completely trained using their own training data. Finally, it supports incremental learning, which we believe is essential for any usable system for activity monitoring. This is because any practical activity monitoring system should be allowed to adapt to slow changes in patterns of activity, brought about by gradual changes in the environment or activities performed.

2.1 Online Clustering Classifier

2.1.1 Learning

To avoid problems in supervised systems, we use an unsupervised clustering algorithm to detect abnormal activities in MAMS. As illustrated in Figure 1, each activity is represented by a data point and activities are grouped into *clusters* based on a *distance* measure. The *centroid* of a cluster represents an average data point amongst all member data points in the cluster. In MAMS, clusters are not labeled because of the unsupervised nature of the learning algorithm.

MAMS constantly monitors a user's activity, as new data points are generated continuously from the sensors. For each new data point, MAMS calculates its distance to each centroid of the existing clusters. If the distance to the closest centroid is smaller than a predefined threshold, then the data point is clustered and merged into this nearest cluster. Otherwise, this activity is considered novel with respect to the current observed activities. MAMS checks with the user whether this novel activity is abnormal. If the user indicates this is actually a normal activity, MAMS forms a new cluster for this activity. Otherwise, if the user confirms something abnormal happened, or if the user fails to respond within a reasonable time interval, MAMS will then quickly contact the home care giver or emergency responder to look into the situation (see Figure 2).

In our implementation we only keep track of the cluster centroid in order to optimize memory usage. In other words, once a new data point is added to a cluster, the cluster centroid is recalculated to incorporate the data point, and the data point is then discarded. Figure 2 illustrates this process in greater detail.

To allow for scalable and continuous learning, the system sets an upper limit for the number of data points stored. Without this bounded system size, real time clustering may become impossible if the number of clusters becomes too large. Therefore, to maintain this mass, a data point is removed from each of the systems *k* clusters for every *k* data points added after the system has reached its specified critical mass. In this fading process, clusters are discarded if their mass eventually fades completely. This effectively means that normal behavior becomes increasingly abnormal if it becomes a less recurrent pattern of activity. On the other hand, an abnormal activity can gradually become normal as its associated cluster grows with more and more observed data points.

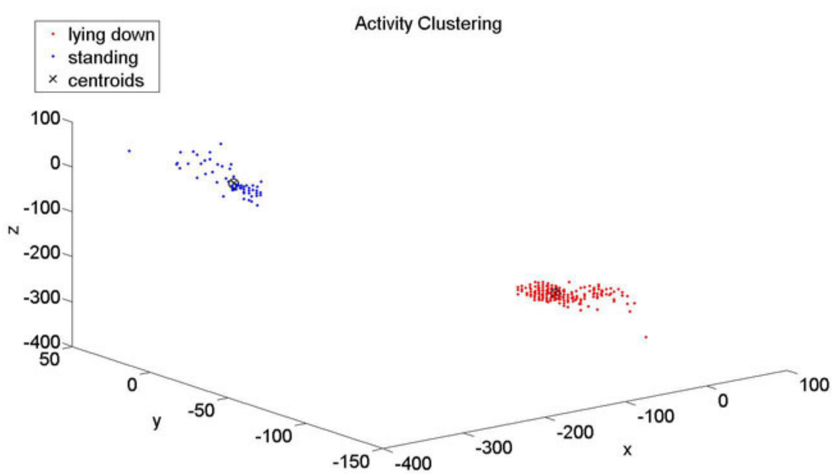

Fig. 1. Similar data points form dense regions or clusters

```
READ data_point a
FOR each existing cluster c_i:
    IF distance (a, c_i ) < min_distance
        min_distance = distance(a, c_i )
        min_cluster = cluster
IF min_distance < clustering_threshold:
     ADD data_point to min_cluster
     RECOMPUTE min_cluster.centroid
ELSE:
     CHECK with user if a is abnormal
     IF 'abnormal' or 'no user response'
        Contact home care worker or emergency
        responders;
     ELSE
        Create new_cluster for a
        Append new_cluster to cluster_array
```

Fig. 2. Pseudo-code for the online clustering process in MAMS

2.2 Activity Feature Vector

The MAMS application considers three characteristics of an activity: posture, movement, and location. Each activity data point is represented by a vector of nine feature values (see Figure 4).

Posture: We compute the mean values of the readings of the three accelerometer axes over a window of 1.8 seconds. In the absence of motion, the mean component is essentially the constant forces of gravity on each of the accelerometer axes.

Movement: We compute the variance of accelerometer axes readings over the same window of 1.8 seconds and relate the variance to the user's current movement.

Location: We use the Wi-Fi based positioning algorithm Redpin [4] to locate a user's indoor position. Redpin uses Wi-Fi fingerprinting to output a user's room level location, e.g. living room, master bedroom, bathroom etc. Fingerprinting is based on the assumption that the Wi-Fi RSS fingerprints seen over time at one location are reasonably stable. Given a database of labeled RSS fingerprints, Redpin uses the K-nearest neighbor (KNN) algorithm to predict the most plausible location of the user. It outputs a single room as the user's current location.

One of the shortcomings of Redpin is that it only outputs a single room where the user might be located. Therefore when computing the distance between two locations, the distance can take on only one of two values: 1 if two locations are from the same room or 0 otherwise. To make the location range smoother, we develop a *pseudo continuous location system* for indoor location. In this system, we consider a user's location as a unit vector in n-dimensional space, where each dimension corresponds to a room in which the user can be. Since this vector is a unit vector, each room component represents the probability of the user being in that room. Figure 3 illustrates the three dimensional case.

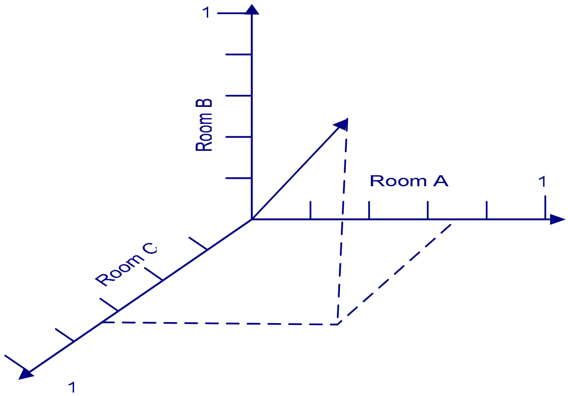

Fig. 3. Room-wise probabilities for three different rooms map a user's location to a point in 3-D space

Accelerometer Mean	Accelerometer Variance	Indoor Location
X = 150	X = 12,150	Room A: 95 %
Y = 120	Y = 5,010	Room B: 3%
Z = 130	Z= 9,800	Room C: 2%

Fig. 4. A sample activity feature vector

2.3 Distance Measurement

As each data point is generated, we attempt to assign it to the closest existing cluster. Closeness is computed as weighted distance of the posture, movement, and location components. For a new data point, *a*, the distance to a cluster centroid, *c*, is calculated using equation 1 below. Here, the net distance is calculated to be the weighted sum of the distances between *a* and *c* for each of the three feature components. The scaling factors assigned to each of the components (α, β, δ) are computed such that, on average, all components contribute equally to the distance measure.

$$\mathrm{dist}(a,c) = \alpha \times \mathrm{posture\_dist} + \beta \times \mathrm{movement\_dist} + \delta \times \mathrm{location\_dist}\,. \tag{1}$$

The posture and movement distances between *a* and *c* are calculated as the Euclidean distances between their accelerometer axes mean and variance vectors respectively. To compute the location distance between the two points, we compute the Euclidean distance between their n-dimensional unit vectors of room-wise probabilities. Equations 2, 3 and 4 capture the formulas for computing the distance between two points for each of the three feature components: posture, movement and location.

$$\text{posture\_dist}(a,c) = \sqrt{\begin{array}{l}(\text{mean}(x_c) - \text{mean}(x_a))^2 + \\ (\text{mean}(y_c) - \text{mean}(y_a))^2 + \\ (\text{mean}(z_c) - \text{mean}(z_a))^2\end{array}} \tag{2}$$

$$\text{movement\_dist}(a,c) = \sqrt{\begin{array}{l}(\text{variance}(x_c) - \text{variance}(x_a))^2 + \\ (\text{variance}(y_c) - \text{variance}(y_a))^2 + \\ (\text{variance}(z_c) - \text{variance}(z_a))^2\end{array}} \tag{3}$$

$$\text{location\_dist}(a,c) = \sqrt{\sum_{\exists \text{room in building}} (\text{prob}_c(\text{room}) - \text{prob}_a(\text{room}))^2} \tag{4}$$

In our study we assume that users holster or place their mobile device in their pockets in a consistent manner. In other words, when the phone is not in use, we assume its orientation with respect to the test subject to be non-changing.

3 Experiments

We test MAMS on 5 test subjects. Test subjects are full time graduate students studying and working in the same building. All students attend the same courses, and for the most part spend their time studying in their cubicles which are all located in the same room.

During the experiments, each test subject was asked to carry a Nokia N95 mobile phone for three days while a logger collected sensor data from the Wi-Fi and accelerometer sensors on each phone. Participants were encouraged to keep the phone with them at all times and to continue using it as they normally would. In other words, they were allowed to continue using the phone for making and receiving phone calls and other functions.

To ensure enough data are collected before the monitoring starts, monitoring only begins after the mass of the system has reached a specified critical level. After this point, both learning and monitoring continue simultaneously. In other words, MAMS does not check with the user if any abnormal activities occur at the beginning of the experiment.

3.1 Anomaly Detection Accuracy

In order to estimate the accuracy of MAMS in detecting normal/abnormal activities, we asked test subjects to perform 5 *abnormal* activities. These activities test various permutations of a user being in an abnormal location, assuming an abnormal posture, or moving around in an abnormal manner. Table 1 shows the list of abnormal activities participants performed and the abnormalities associated with each activity.

In this experiment a mobile application was installed on each subject's mobile phone. This application requests test subjects to perform a series of tasks. At random times, the device plays a distinct beep sound to instruct the test subject to start performing the abnormal task. Tasks were 10 seconds in length, and during this time interval, the subjects were expected to continue performing the abnormal activity. The purpose of this was to clearly mark the beginning and end of an abnormal task for evaluation.

Table 1. Abnormal Activities

	Abnormal Location	Abnormal Movement	Abnormal Posture
Jumping Jacks Student Cubicle		✓	✓
Lying Down Next to Student Cubicle			✓
Seated in Conference Room	✓		
Lying down in conference room	✓		✓
Jumping Jacks Conference Room	✓	✓	✓

We study how the system's precision and recall vary with respect to two parameters in MAMS: the clustering threshold and the behavior model size. We define precision and recall as follows.

$$\text{precision} = \frac{\text{Number of labeled abnormal activities correctly classified by MAMS}}{\text{Number of abnormal activities classified by MAMS}} \quad (5)$$

$$\text{recall} = \frac{\text{Number of labeled abnormal activities correctly classified by MAMS}}{\text{Total number of labeled abnormal activities}} \quad (6)$$

The clustering threshold is the maximum allowable distance a data point can be from its closest cluster centroid for it to be successfully assigned to that cluster. Data points further than this threshold are considered to pertain to novel activities. In the context of anomaly detection, novel activities are flagged as abnormal ones. Figure 5 shows the precision, recall and F1 score (harmonic mean of precision and recall) as the clustering threshold is varied. In this experiment, we use 2 days of unlabeled normal activity logs to generate the behavior models. We test the system's precision and recall using the third day of normal activity logs for each user and their 5 labeled abnormal activity logs.

As shown in Figure 5, increasing the threshold in the very beginning increases the F1 score, since the precision value is increasing at a larger rate than the recall is decreasing. However, after a certain point, the rate of increase in precision is exceeded by the drop in recall as the threshold is further increased.

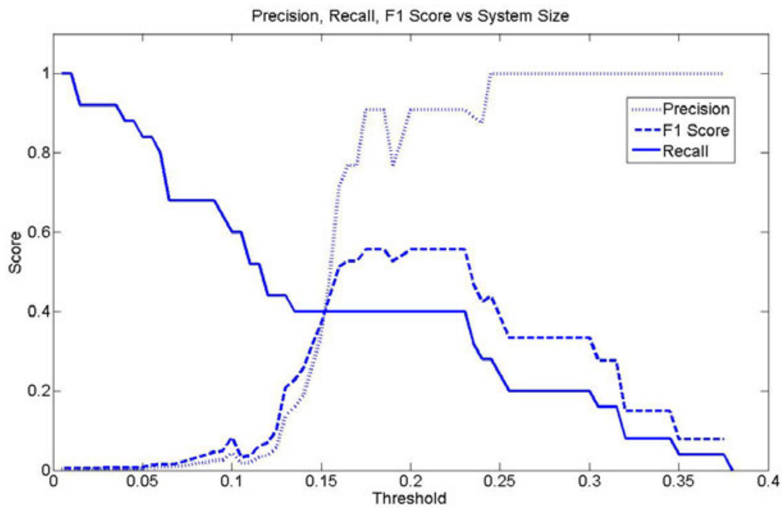

Fig. 5. Precision, recall, and F1 score vs. clustering threshold

Figure 6 illustrates the effects of increasing the model size on cases of reported novel activities. The behavior model size determines the number of historical data points kept in the MAMS system among all clusters. In Figure 6, model size is represented as log-hours, which is a measure of the time it takes the system to reach its critical mass. As the graph shows, at the early stage, MAMS reports a lot of "novel" activities since the behavior model has not covered enough normal activities yet. As the system mass increases, i.e. more user behaviors have been observed by the system, MAMS reports fewer activities to be "novel" or "abnormal". At around 4 log-hours, the system's sensitivity to abnormal data points levels off. Choosing a model

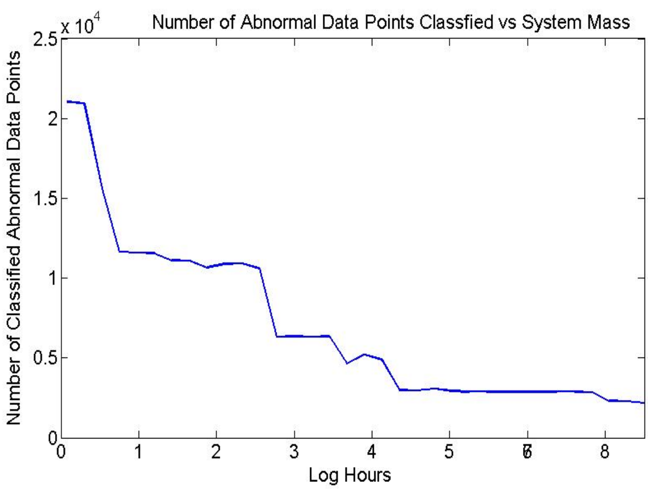

Fig. 6. As the system mass increases, the number of classified abnormal data points decreases

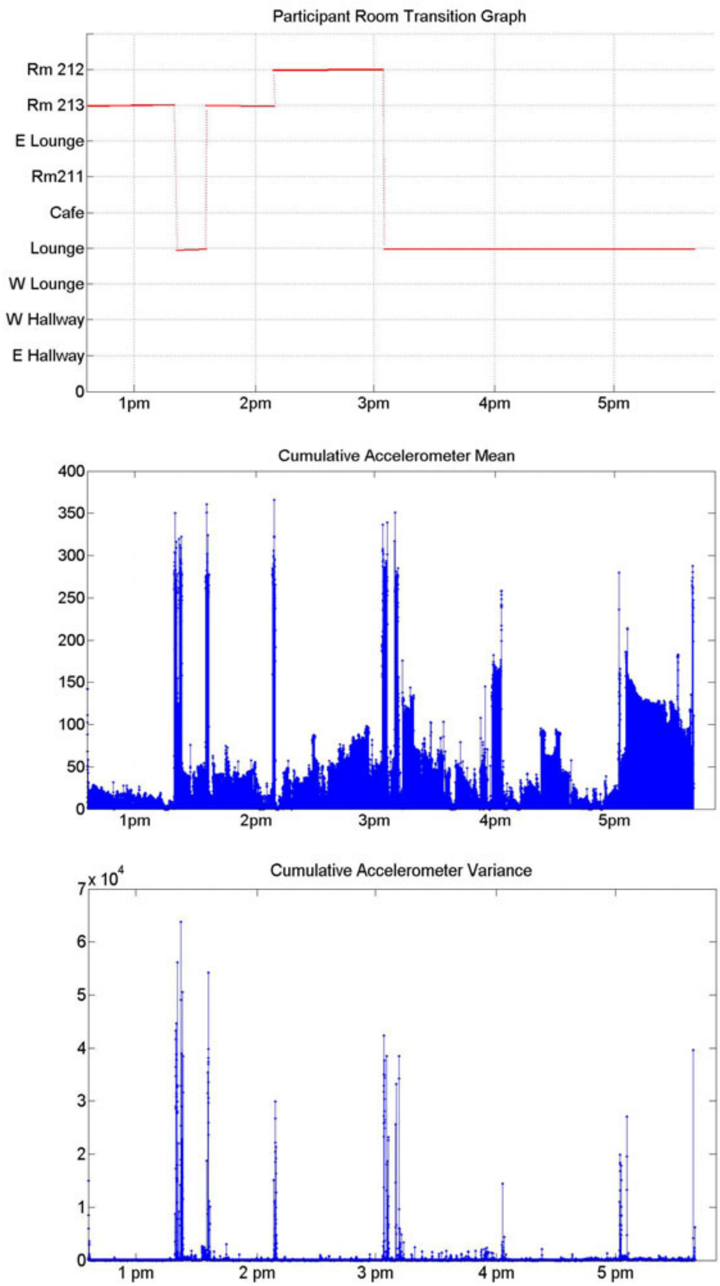

Fig. 7. Correlation among location, posture, and movement of a subject over a 5 hour period

size larger than 4 log-hours yields only marginal improvements in the decrease in false positives when identifying anomalous activities. Ultimately, choosing an appropriate model size is a trade-off between the performance constraints of the mobile device, and the level of false positives. Further experiments will be needed to determine the real-time performance of the system as the model size is increased.

Choosing a model size of 4 hours and the optimal clustering threshold, MAMS achieves 90% precision and 40% recall.

The three graphs in Figure 7 show the correlation between a user's location, posture and movement. The room location is indicated in the uppermost room transition graph. Also shown are figures illustrating the sum of the means and variance values of the three accelerometer axes. The results show that transitions between rooms bring about temporally aligned abrupt changes in the accelerometer mean and variance. However, while the user is within a room, both the accelerometer mean and variance fluctuate very little. These small fluctuations may be due either to jitter in the axes readings or to very small movements subjects make while seated or standing. This result supports our hypothesis that activities are location-dependent.

We find that removing the location feature has little effect on the system performance in our experiments (Table 2). This may be due to the fact that student participants spend most of their time seated in classrooms and at their cubicles. Future experiments may be needed to test the performance of the location feature on elderly subjects in their home environments.

Table 2. MAMS performance with the location feature enabled and disabled

Active MAMS Features	Precision	Recall	F1
Movement, Posture, Location	0.901	0.400	0.556
Movement, Posture	0.909	0.400	0.556

3.2 Cross-User Model Portability

MAMS learns behavior models for each individual user by monitoring his/her activities. We conducted experiments to test how well one user's behavior model works for other users, i.e., the portability of behavior model across different users.

Table 3 shows the results of cross-user model portability experiments. Each row in Table 3 represents the behavior model of a particular user and the columns are test subjects. Behavior models are learned from normal activity data collected over a period of 2 days. We use normal activity data from a third day as testing data. The entries in each row represent the percentage of activities which were recognized as repeated activities. With the exception of one user (user 4), MAMS performs best when the behavior model is trained from the same user to which it is being applied. These results imply that users tend to exhibit repeated patterns of activities which are unique from one person to another, and this underpins our approach of learning user-specific activities for activity recognition.

Table 3. Normal activity recall rate for behavior models applied to different users

	User1	User2	User3	User4	User5
User1 Cluster	**0.537**	0.336	0.030	0.489	0.142
User2 Cluster	0.648	**0.710**	0.229	0.463	0.229
User3 Cluster	0.422	0.390	**0.631**	0.356	0.562
User4 Cluster	0.204	0.270	0.033	0.350	**0.672**
User5 Cluster	0.171	0.124	0.000	0.110	**0.642**

4 Related Work

Much research into context and activity classification using accelerometers focuses on learning a predefined set of activities. Fabian et al. make use of wearable devices consisting of a gyroscope and a tri-axial accelerometer [5]. Three of these devices are placed on a subject's hip, dominant ankle and wrist. Using intensity values from the accelerometer axes as features, and feed forward neural networks, they report a 79.76% recognition rate for a set of 7 activities. Bao and Intille, experiment with various classifiers to classify various static and dynamic activities from accelerometer data [6]. They conclude that decision trees, which yielded 89.3% accuracy rates, perform the best. Similar studies with labeled activity recognition using accelerometer data also report accuracies in the 80%-90% range [7][8][9]. Although these approaches typically perform well for a small set of activities, they generally don't span the entire set of activities a person performs. Hence they are of limited use in real-time monitoring applications.

Studies have shown that unsupervised learning can indeed be applied to activity recognition with limited success. Krause et al. present an online learning algorithm which uses self-organizing maps and clustering to identify a set of unlabelled activities [10]. Transient or infrequent activities are filtered out to leave a set of activities which are considered to be the most recurring ones. The disadvantage of this scheme is that the number of contexts or activities generated is less than the number of activities participants identified. Hein and Kirste present a hybrid k-means clustering and Hidden Markov Model (HMM) approach to detect high-level activities [11]. Applying their classifier to data from a tri-axial accelerometer, gyroscope, and magnetometer, they report 75% recognition accuracy for high level unlabeled activities. Nguyen et al. present two unsupervised event segmentation techniques: coherent event clustering and unusual event clustering [12]. Coherent event clustering iteratively segments the raw accelerometer data into sequences and coherently clusters the segments into event classes. In unusual event clustering, one starts with a usual event model and iteratively updates the usual event model based on outlier events identified by the current model. Both methods model the event/activity using HMM and use Gaussian Mixture Models to represent the emission between the state transitions. Such models are quite complex and non-trivial to train. The large number of parameters needs to be estimated by iterative Expectation Maximization training, making it difficult to use such algorithms on mobile devices especially when online training is desired.

5 Conclusion

We describe the MAMS system, an activity monitoring system that classifies abnormal activity based on deviations from learned normal behavior. We find that our system can recognize anomalous activities based on different permutations of novel location, posture, or movement. This system learns incrementally and adapts to slow changes in activity patterns. As a result, recurrent patterns of activity gradually become normal, while less recurrent ones become anomalous. On this basis, we believe this system can be used on mobile phones to monitor the homebound elderly. On a short time scale, it may be used to detect critical events such as falls, while on a longer time scale, the system can be used to track changes in quality of life based on increasing or decreasing levels of anomalous activities.

Our results show that generated activity clusters tend to be user specific. This strengthens our approach of generating user-specific models as opposed to using one model for all users. The system recognizes abnormal activities with 90% precision and detects 40% of abnormal activities. This low recall rate may be due to the small dataset size and the pilot nature of our study. In future studies we plan to test with a wider range of labeled abnormal activities and use more training days and users. Additionally, we will evaluate several other classifiers to find one which works best for the purpose of real-time anomalous activity detection.

Although our results show that the location feature appears to bear little effect on the performance of our anomaly classifier, we will re-evaluate the performance of this feature in a larger study.

Acknowledgements

This research was supported by a grant from Nokia Research center, by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office, and by Panasonic. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies.

We also acknowledge the efforts of Patricia Collins, Pei Zhang and the anonymous referees for their valuable feedback and suggestions on earlier drafts of this paper.

References

1. Projections of the Population by Age and Sex for the United States: 2010 to 2050, `http://www.census.gov/population/www/projections/summarytables.html`
2. Western Maine Community Action, Keeping Seniors Home, `http://www.wmca.org/Keeping_seniors_home.htm`

3. Falls Among Older Adults: Summary of Research Findings, http://www.cdc.gov/ncipc/pub-res/toolkit/SummaryOfFalls.htm
4. Bolliger, P.: Redpin - adaptive, zero-configuration indoor localization through user collaboration. In: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, pp. 55–60. ACM, New York (2008)
5. Fabian, A., Gyorbiro, N., Homanyi, G.: Activity recognition system for mobile phones using the MotionBand device. In: Proceedings of the 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, Article No. 41, ICST, Brussels Belgium (2008)
6. Bao, L., Intille, S.: Activity Recognition from User-Annotated Acceleration Data. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
7. Aminian, K., Robert, P., Buchser, E.E., Rutschmann, B., Hayoz, D., Depairon, M.: Physical activity monitoring based on accelerometry: validation and comparison with video observation. Identification of Common Molecular Subsequences 37(3), 304–308 (1999)
8. Mantyjarvi, J., Himberg, J., Seppanen, T.: Recognizing Human Motion with Multiple Acceleration Sensors. In: 2001 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, vol. 2, pp. 747–752. IEEE, Los Alamitos (2001)
9. Randell, C., Muller, H.: Context Awareness by Analysing Accelerometer Data. In: The Fourth International Symposium on Wearable Computers, Atlanta, GA, pp. 175–176. IEEE, Los Alamitos (2000)
10. Krause, A., Sieworik, D., Smailagic, A., Farringdon, J.: Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing. In: Proceedings of the 7th IEEE International Symposium on Wearable Computers, p. 88. IEEE Computer Society, Washington (2003)
11. Hein A., Kirste T.: Towards Recognizing Abstract Activities: An Unsupervised Approach. In: Proceedings of the 2nd Workshop on Behaviour Monitoring and Interpretation. pp 102-114. Universitat Bremen, Bremen Germany (2008).
12. Nguyen, A., Moore, D., McCowan, I.: Unsupervised Clustering of Free-Living Human Activities using Ambulatory Accelerometry. Engineering in Medicine and Biology Society 22, 4895–4898 (2007)