Toward Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones

Jun Yang
Nokia Research Center
Palo Alto, CA 94304, USA
jun.8.yang@nokia.com

ABSTRACT

In this paper, we perform physical motion recognition using mobile phones with built-in accelerometer sensors. Sensor data processing and smoothing techniques are discussed first to reduce the special noise present in phone-collected accelerometer data. We explore orientation-independent features extracted from vertical and horizonal components in acceleration as well as magnitudes of acceleration for six common physical activities, such as sitting, standing, walking, running, driving and bicycling. We find decision tree achieves the best performance among four commonly used static classifiers, while vertical and horizonal features have better recognition accuracy than magnitude features. Furthermore, a well-pruned decision tree with simple time domain features and less over-fitting on the training data can provide a usable model for inferencing a physical activity diary, refined by a similarity match from k-means clustering results and smoothed by an HMM-based Viterbi algorithm.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: Classifier design and evaluation, Feature evaluation and selection, Pattern analysis

General Terms

Algorithms, Design, Experimentation, Measurement, Performance

Keywords

Activity Recognition, Motion Sensors, Mobile Phones, Decision Tree, K-Means, Hidden Markov Model (HMM)

1. INTRODUCTION

User context awareness is one of the emerging mobile applications and services in the area of ubiquitous computing. In general, user context means user's activity, location, preference, situation, emotion, etc. With increasingly powerful mobile phones, most of user contexts include a variety of sensing components embedded in mobile phones, such as accelerometer, GPS, microphone, Bluetooth, camera, etc. In essence, mobile phones can create mobile sensor networks that are able to collect sensor data important to a user's daily life, namely, what is the user doing, where is the user, and whom is the user staying with? In this paper, we investigate possibilities and feasibilities of a context awareness service that monitors users' daily physical activity with portable mobile phones. The same machine learning method can be extended to the other domains, such as location awareness and proximity detection.

Some existing works have explored user activity inferencing methods with accelerometer sensors [15, 4, 12]. They can be divided into two major approaches: sensor-worn lab experiment approach and sensor-enabled mobile phone approach. In a well-cited work, Bao and Intille [4] first used multiple accelerometer sensors worn on different parts of a human body to detect common activities such as sitting, standing, walking or running. Lester et al [12] developed a small low-power sensor board to be mounted on a single location on the body. Then a hybrid approach was presented to recognize activities, which combines the boosting algorithm to discriminatively select useful features and HMMs to capture the temporal regularities and smoothness of activities. Furthermore, Lester et al extended their work to a practical personal activity recognition system in another paper [11]. However, the assumption made from lab experiment is usually not suitable for practical applications in regular mobile environment. In [13], a phone-centric sensing system, CenceMe, it is assumed that mobile phone's position on the human body is fixed (e.g. in a pocket, clipped to a belt or on a lanyard), and an inference model can be trained according to a handful of body positions. However, one main issue is mobile's phones could be placed at various locations to each individual habit, plus different orientations are highly possible in each of the positions. For example, in a pocket, phone's posture can be upside or upside down, while its screen can be facing inside or facing outside. It is not fully convinced the model without considering phone's orientation can be applied to a large group of users.

There are already previous works considering phone's orientation problem. In [10], a method is described to derive the body part location of an accelerometer sensor where the user is walking. Also in [7], an adaptive context inference scheme can automatically detects the sensor position on the user's body and selects the most relevant inference method dynamically. However, it is not practical to divide all kinds

IMCE'09, October 23, 2009, Beijing, China.

of phone orientations into a handful categories and make an inference model according to each of them.

In [14], Mizell has shown the accelerometer signal averages over a reasonable time period can produce a good estimate of the gravity-related component. In a previous work [8], it has shown that many activities are determined by vertical orientation and changes thereof. In a most recent paper [9] dealing with sensor displacement in motion-based onbody activity recognition systems, the authors have discussed if only an accelerometer is available, the best we can do is to identify the segments of the signal dominated by the gravity component and make recognition based on the information abut vertical orientation. Incited by these results, we use orientation-independent features, such as vertical and horizontal components in acceleration as well as magnitude of acceleration to recognize daily motion activities. As far as we now, this is a first work to explore the feasibility of orientation-independent features for physical activity recognition. We further discuss how to reduce the complexity of feature computation for mobile devices with less computing power while maintaining good recognition performance. The work can be generalized to a large scale of context awareness applications using phone-based physical activity recognition.

In the rest of this paper, data collection is briefly introduced in Section 2. Section 3 presents a smoothing technique to reduce the noise in accelerometer data. In Section 4, we explain how to extract orientation-independent signals and compute associated features. Section 5 presents different recognition methods and experiment results for six common physical activities. In Section 6, a decision model from Section 5 is used to generate a physical activity diary, followed by refining from clustering results and smoothing from a second-level HMM. Finally, we conclude the paper in Section 7 and acknowledge is made in Section 8.

2. DATA COLLECTION

We use Nokia N95 phones to collect accelerometer data. The N95 phone is equipped with a built-in accelerometer that is an tri-axial MEMS motion sensor (LIS302DL) made by STMicro. It has dynamically user selectable full scales of $\pm 2g/\pm 8g$ (where g is the gravitational acceleration, $g = 9.81m/s^2$) and it is capable of measuring accelerations with an output data rate of 100 Hz or 400 Hz. The digital output has 8-bit representation with each bit equaling to $18mg$. The configuration of sensor device on N95 phones is set to $\pm 2g$. The sampling frequency of N95 accelerometer sensor is reduced to approximately 36 Hz by calling the Nokia Python S60 sensor module over Symbian platform. The N95 accelerometer sensor was originally only used for video stabilization and photo orientation, in order to keep landscape or portrait shots oriented as taken. Nokia Research Center has allowed an application interface directly to accelerometer, allowing software to use the data from it. Nokia has released a step counter application [3] to demonstrate this. Another Nokia-created application taking advantage of the accelerometer is Nokia Sports Tracker [2].

We collected data for everyday activities that involve normal body movement and range in level of intensity, such as sitting, standing, walking, running, driving and bicycling. The activities were performed by four subjects over one afternoon. An efficient data collection client running on Nokia S60 mobile phones, Nokia Simple Context [1], is used to collect every 10 seconds of accelerometer samples and upload the data to a server. Accelerometer data between the start and end times were labeled with the name of that activity. Using the same data collection tool, we can collect a large set of unlabeled accelerometer data of one person over several months.

For some experiment, a Python script interacting with Nokia Python S60 sensor module is used to collect instant accelerometer data from N95 phones directly. The sampling duration and frequency can be set by different duty cycles. The data is labeled by pre-input activity labels in the script.

3. DATA PROCESSING

Each reading of accelerometer sensor consists 3-D accelerations along X-axis, Y-axis and Z-axis according to local coordinate system of *current* phone orientation. For example, a few seconds of samples of $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ readings from a phone sitting on a table are plotted in Figure 1. It can be seen that both $\boldsymbol{x}$ and $\boldsymbol{y}$ are around -15 while $\boldsymbol{z}$ is around -280. Theoretically x and y should be zero since there is no acceleration at all, and z should be measured as $-g$. So there exists sensor reading errors that affect data quality. As shown in the top subplot of Figure 1, there is a ± 5 jittering noise present in accelerometer data when phone is fully static on the table. We can reduce the effect of jittering noise by scaling down $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ readings by a factor M and rounding, followed by a smoothing technique using a moving-average filter of span L, i.e.,

$$(\boldsymbol{x}^{'}, \boldsymbol{y}^{'}, \boldsymbol{z}^{'}) = \text{mv-filter}\left(\text{round}\left[(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})/M\right], L\right). \quad (1)$$

After selecting $M = 5$ and $L = 5$, a new series of readings

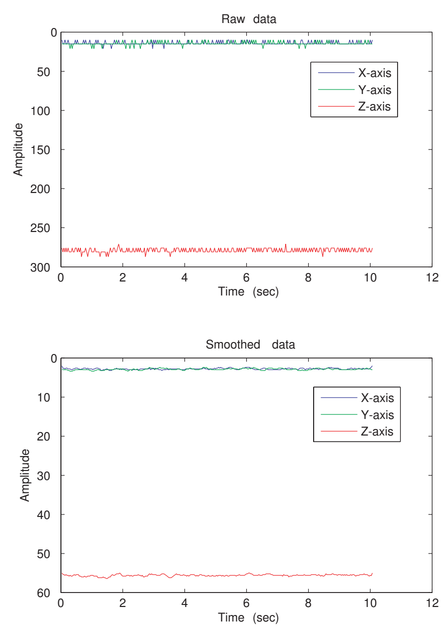

Figure 1: Accelerometer readings from a static phone

are generated and shown in the bottom subplot of Figure 1. Statistics of raw accelerometer readings, such as mean and standard deviation, are

$$(m_{\mathbf{X}}, m_{\mathbf{Y}}, m_{\mathbf{Z}}) = (-13.97, -14.53, -279.01)$$

and

$$(\sigma_{\mathbf{X}}, \sigma_{\mathbf{Y}}, \sigma_{\mathbf{Z}}) = (2.23, 2.23, 2.85),$$

while mean and standard deviation of smoothed accelerometer readings are

$$(m_{\mathbf{X}'}, m_{\mathbf{Y}'}, m_{\mathbf{Z}'}) = (-2.78, -2.89, -55.59)$$

and

$$(\sigma_{\mathbf{X}'}, \sigma_{\mathbf{Y}'}, \sigma_{\mathbf{Z}'}) = (0.18, 0.20, 0.24).$$

It can be seen that proposed smoothing technique greatly removes jittering noise and smooths out data by more significant standard deviation reduction than mean reduction. The effect of proposed smoothing effect on accelerometer readings from a moving phone is shown in Figure 2. The de-noising process is important for building a recognition model using mean and stand deviation.

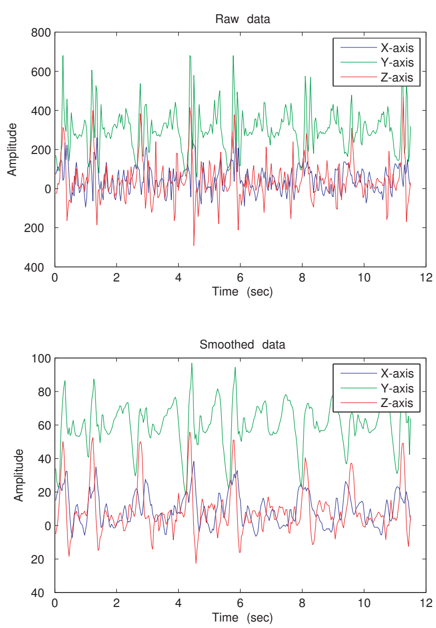

Figure 2: Accelerometer readings from a moving phone

4. FEATURE EXTRACTION

As we know from tri-axial accelerometer sensor mechanism, sensor orientation determines the local coordinate system of each (x, y, z) reading. Most previous work on activity recognition used body-worn accelerometer sensors and attached them to the body in a known position and orientation relative to the body. When mobile phones are used as portable sensors, the assumption will be no longer valid. As illustrated in Figure 3, accelerometer readings from phones of four people A,B,C and D sitting are plotted in four subplots, respectively. Although fours phones are all located in a pocket and positive Z-axis values indicate all four phones are almost facing downward, different orientations cause z readings varying from person to person. It can be seen that (x, y) readings of one person are significantly different from the other person.

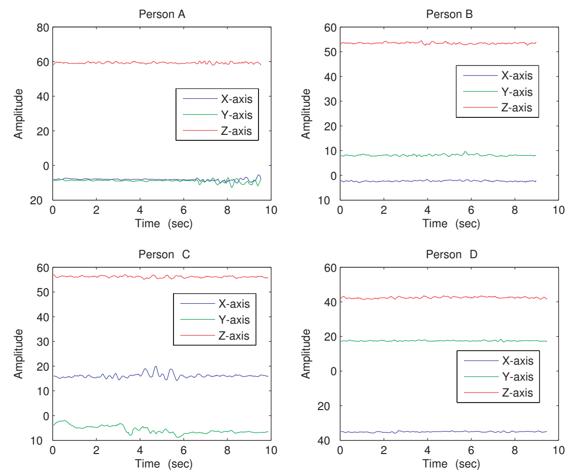

Figure 3: Processed accelerometer data from four sitting phones

One easy solution to avoid orientation problem is using magnitude of each (x, y, z) accelerometer signal. However, this kind of operation will cause acceleration loss 3-d directional information. In [14], Mizell has shown the signal average on each axis over a reasonable time period can produce a good estimate of the gravity-related component. We take a similar approach here to estimate the gravity component from each segment of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ readings. Our estimation internal is set to the same as sample duration, which is usually 10 seconds and long enough to estimate the vertical acceleration vector $\bar{v}$ corresponding to gravity. Therefore, $\bar{v} = (m_{\mathbf{X}}, m_{\mathbf{Y}}, m_{\mathbf{Z}})$, where $m_{\mathbf{X}}, m_{\mathbf{Y}}$ and $m_{\mathbf{Z}}$ are means of respective axes for the sampling period. $\bar{v}$ is normalized to $\bar{v}_{norm}$. Let $\bar{a}_i = (x_i, y_i, z_i), i = 1, 2, \ldots, N$, be the vector at a given point in the sampling interval, where N is the length of sample points in the sample duration. The projection of $\bar{a}_i$ onto the vertical axis $\bar{v}_{norm}$ can be computed as the vertical component inside $\bar{a}_i$. Let p_i^{in} be the inner product and $\bar{p}_i$ be the projection vector, i.e.,

$$p_i^{in} =< \bar{a}_i, \bar{v}_{norm} >, \quad \bar{p}_i = p_i^{in} \cdot \bar{v}_{norm},$$

then the horizontal component $\bar{h}_i$ of the acceleration vector $\bar{a}_i$ can be computed as vector subtraction, i.e.,

$$\bar{h}_i = \bar{a}_i - \bar{p}_i.$$

However, it is impossible to know the direction of $\bar{h}_i$ relative to the horizontal axis in global 3-axis coordinate system. We only know $\bar{h}_i$ lies in the horizontal plane which is orthogonal to estimated gravity vector $\bar{v}$. So we simply take the magnitude of $\bar{h}_i$, denoted by $\|\bar{h}_i\|$, as a measure of horizontal movement. The results of above algorithm generate two waveforms of $\{p_i^{in}, i = 1, 2, \ldots, N\}$ and $\{\|\bar{h}_i\|, i = 1, 2, \ldots, N\}$, which are amplitude of the vertical components (positive or negative) and magnitude of the horizontal components (only positive), respectively. Each waveform is almost independent of orientation of the phone taking instant accelerometer samples.

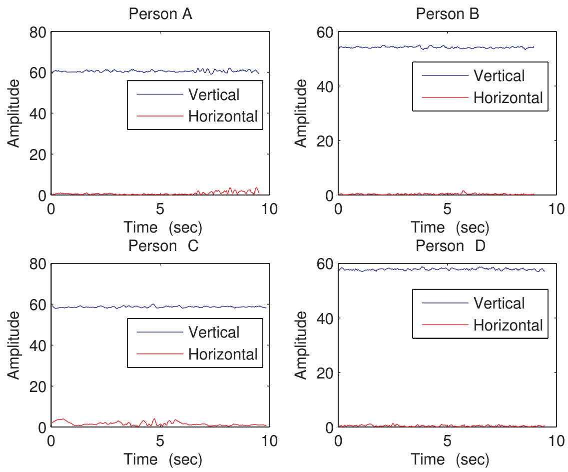

Figure 4: Vertical and horizonal components extracted from accelerometer readings in Figure 3

As illustrated in Figure 4, the vertical and horizontal components of accelerometer data in Figure 3 are plotted according to above algorithm accordingly. All four persons have little physical movement along Z-axis, which reflects the dynamic range of vertical components is not significant. Person C has slight physical movements along X-axis and Y-axis when he is sitting. It reflects the horizontal component of his accelerometer data has more significant fluctuations than those of Person A, B and C.

After the vertical and horizonal components are extracted from the smoothed accelerometer data, features are computed on each pair of $\{p_i^{in}, i = 1, 2, \ldots, N\}$ and $\{\|\bar{h}_i\|, i = 1, 2, \ldots, N\}$. A window size of $N^{'} = 2^{\lceil log_2 N \rceil}$ samples enables a fast computation of FFTs used for some of the features. The following time-domain features are computed: mean, standard deviation, zero crossing rate, 75% percentile and interquartile range. For these two waveforms, the zero crossing level is set to their mean and all five features are computed. Since our accelerometer data has a lower sampling frequency at 36 Hz, the signal bandwidth is around 18 Hz that doesn't include enough higher frequency information above 10 Hz. We compute the power spectrum centroid and frequency-domain entropy of the vertical and horizontal components. The cross-correlation between $\{|p_i^{in}|, i = 1, 2, \ldots, N\}$ and $\{\|\bar{h}_i\|, i = 1, 2, \ldots, N\}$ is computed as well. There are 15 features in total, named as **V/H** feature set

- **Vertical features**: meanV, stdV, zcrV, 75PercentileV, InterQuartileV, SpecCenV, entropyV;
- **Horizonal features**: meanH, stdH, zcrH, 75PercentileH, InterQuartileH, SpecCenH, entropyH;
- **Cross-correlation feature**: Corre

The same features are also computed from the waveform of magnitude $\{m_i = \|\bar{a}_i\|, i = 1, 2, \ldots, N\}$ (for comparison purpose), named as **M** feature set

- **Magnitude features**: meanM, stdM, zcrM, 75PercentileM, InterQuartileM, SpecCenM, entropyM.

In order to save computational power on the phone, we select distinctive features from the feature set using the correlation based feature selection method from the WEKA toolkit. The new feature subset should contain highly correlated features within the particular class but are uncorrelated with each other. We add one more rule that less frequency features should be used as possible as we can, since FFT transformation involves more computing power than pure computation in time domain. The feature selection process will be interpreted in the following section.

5. MOTION RECOGNITION AND EXPERIMENT RESULTS

Accelerometer data were collected from four persons during one afternoon time. Each person performed six daily activities such as sitting, standing, walking, running, driving and bicycling from 10 minutes to half an hour. There are total 908 samples collected from three persons as training data, which are listed in Table 1. The other 245 samples collected by the rest person will be used as test data set. Each sample's duration is around 10 seconds.

Table 1: Accelerometer training data set

Index	Label	Number of samples
a	Sitting	138
b	Standing	146
c	Walking	121
d	Running	64
e	Driving	330
f	Bicycling	109

We evaluate and compare several classifiers provided by WEKA, namely C4.5 Decision Trees (DT), Naive Bayes (NB), k-Nearest Neighbor (kNN) and the Support Vector Machine (LibSVM) [5]. 10-fold cross-validation is used for testing. Different feature subsets are listed in Table 2, where the **V/H** feature set and **M** feature set are separated by double line. In **V/H** feature set, WEKA selected features are selected by WEKA AttributeSelection filters, including {meanV, stdV, 75PercentileV, InterQuartileV, SpecCenV, entropyV, meanH, entropyH, Corre} and simplified features are refined from WEKA selected features by removing frequency features and replacing them with StdH, including {meanV, stdV, 75PercentileV, InterQuartileV, meanH, stdH, Corre}. TD features means time-domain only features. In **M** feature set, selected time-domain features are {meanM, stdM, 75PercentileM, InterQuartileM}. From the accuracy results, we conclude that vertical features contain more motion information than horizonal features, and time-domain features are sufficient for different daily physical activities. Decision trees is found to achieve high recognition accuracy with acceptable computational complexity. kNN and SVM can achieve good performance for selected time-domain magnitude features.

As to the decision tree classifier, the performance of **V/M** feature set outperforms that of **M** feature set. One confusion matrix obtained from DT classifier using simplified V/H features is put in Table 3, and the other confusion matrix obtained from DT classifier using selected time-domain

Table 2: **Feature subsets and recognition accuracy**

Feature Set	DT	NB	kNN	SVM
All V/H features	90.6%	68.7%	77.9%	84.3%
Vertical features	88.7%	68.3%	85.0%	85.1%
Horizonal features	71.0%	66.3%	68.0%	69.3%
TD features	89.8%	68.6%	78.1%	84.8%
Weka-S features	90.4%	68.1%	79.7%	86.3%
Simplified features	90.2%	67.8%	84.6%	86.0%
Magnitude features	88.1%	75.3%	86.8%	86.0%
Selected TD features	88.5%	74.6%	89.2%	89.1%

magnitude features is put in Table 4. Vertical and horizonal features can better distinguish sitting, driving and bicycling motions compared to magnitude features. It is because horizonal feature can detect movement better than magnitude feature.

Table 3: Confusion matrix for DT classifier using simplified V/H features

Labeled activity	Recognized activity a	b	c	d	e	f
a	120	4	0	3	11	0
b	1	124	1	0	17	3
c	0	0	111	0	5	5
d	2	0	3	59	0	0
e	11	9	3	1	305	1
f	0	2	1	1	5	100

Table 4: Confusion matrix for DT classifier using selected TD magnitude features

Labeled activity	Recognized activity a	b	c	d	e	f
a	116	7	0	0	15	0
b	6	124	0	0	16	0
c	0	0	113	1	5	2
d	0	0	4	60	0	0
e	18	5	1	0	299	7
f	0	0	2	0	15	92

We further consider using some easily-computing features, such as mean and standard deviation, that can be implemented on the phone with minimal computational complexity. For an C4.5 decision tree model, When only {meanV, stdV, meanH, stdH} features are used, the accuracy is 87.6%. When only {meanM, stdM} features are used, the accuracy is 85.7%. Compared to the accuracy results in Tabe 2, there is at most 3% performance degradation, which is affordable due to a significant reduction of feature computation. We call them **simple** features. The generated decision trees can be further pruned according to the error rate of 10-fold cross-validation. The tradeoff between error rate and tree size is shown in Figure 5 and a well-pruned decision tree model (Tree 1) generated from simple V/H features is presented in Figure 6 with reduced accuracy 80.7%. For simple magnitude features, the tradeoff between error rate and tree size is shown in Figure 7 and a similar pruned decision tree model (Tree 2) are shown in Figure 8 with reduced accuracy 80.3%. The pruned tree model has less over-fitting effect on the training data set we expect for testing.

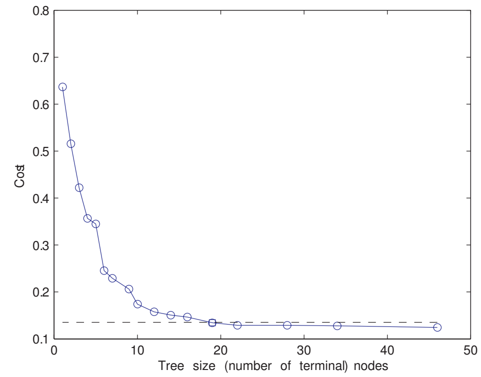

Figure 5: Tree size vs. error rate

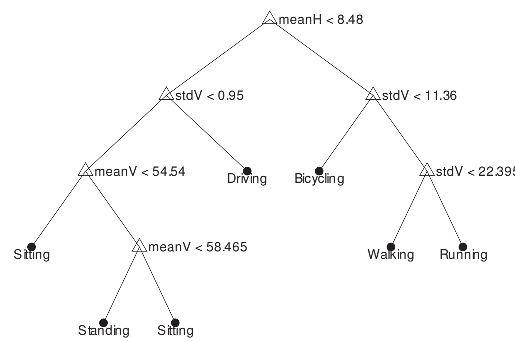

Figure 6: Tree 1 generated from simple V/H features

After carefully observing the decision tree models in Figure 6 and Figure 8, we find they are quite similar to each other except that the root node in Tree 1 is meanH while it is stdM in Tree 2. For the other tree nodes, the vertical pair (meanV, stdV) in Tree 1 is replaced by the magnitude pair (meanM, stdM) in Tree 2. It is understandable the mean of horizonal acceleration can distinguish sitting, standing and driving from walking, running and bicycling as well as the standard deviation of magnitude of acceleration. In Tree 2, small standard deviation means sitting, standing and driving while smaller standard deviation means sitting and standing. And the mean is used to distinguish sitting from standing in Tree 2 because of the following reasons: 1. Accelerometer sensor readings from different phones have different offsets in X, Y, Z-axis when phones are sitting, which cause the value of meanM to vary from phone to phone; 2. Accelerometer sensor readings from the same phone on different orientations have slightly different magnitudes of acceleration, which cause meanM of sitting slightly different from meanM of standing. The decision tree model actually uses this change to distinguish sitting from standing, as shown in Tree 2.

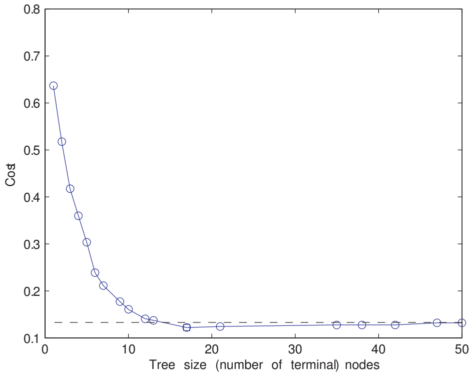

Figure 7: Tree size vs. error rate

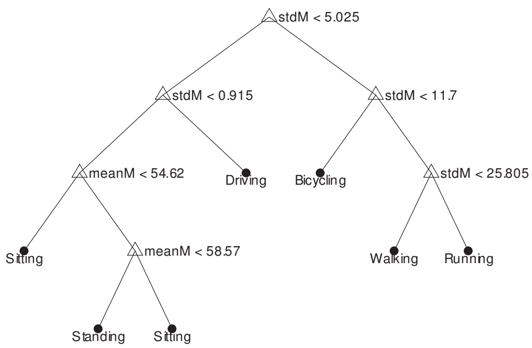

Figure 8: Tree 2 generated from simple magnitude features

After applying the above two simple tree models to the test data set, the accuracy is 66% for Tree 1 and 69% for Tree 2. From our experience, sufficient training data set with adaptive calibration on each phone's accelerometer readings will improve the classification accuracy between sitting and standing.

6. TOWARD PHYSICAL ACTIVITY DIARY

Our goal in this paper is to use phone orientation independent features to generate a general motion recognition model and inference user's physical activity diary by this model. Unlabeled accelerometer data set is collected from one person over several months. This person is a professional researcher with most time sitting and one hour driving every day. The data is collected almost 24 hours every day. The accelerometer sensor is sampled every 4 minutes for 10 second sample duration and the data is uploaded to a server with time stamps for storage. One full week data is selected out of the data set and there are totally 4098 samples there. After the two simplified decision tree models introduced in Section 5 are applied to the data, the following two pie charts are obtained in Figure 9 and Figure 10, respectively. The distribution of six common activities in both figures are quite similar to each other, except Tree 1 classifies a little more bicycling and running, a little less driving than Tree 2. Are they really reflecting the percentage of the person's one week physical activities? Are the six activities labels meaningful enough? This is the question we try to explore in the section.

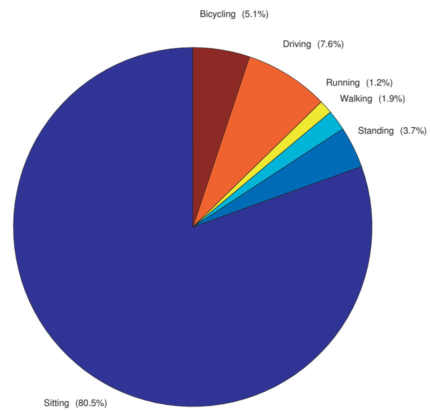

Figure 9: Pie chart of physical activities classified by Tree 1

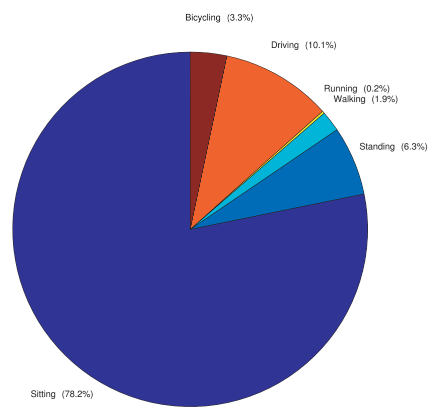

Figure 10: Pie chart of physical activities classified by Tree 2

For the sake of space, we use simple magnitude features (meanM, stdM) to investigate this problem. The same method can be applied to simple vertical and horizontal features. First we plot the classified magnitude features by Tree 2 with different colors in Figure 11.

Then we take a look at the distribution of (meanM, stdM) of 4098 samples and cluster them into six clusters by k-means method. The results of six cluster centroids and percentages are listed in Table 5 with colored clusters shown in Figure 12. The percentage proportion of these six clusters

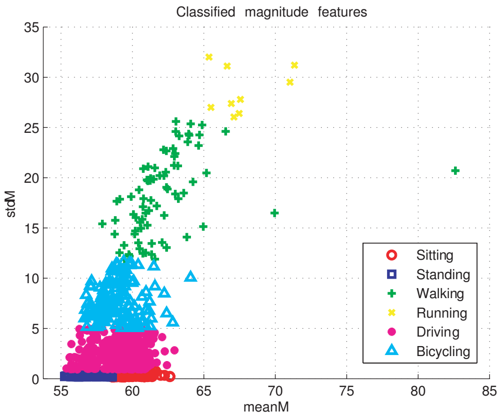

Figure 11: Map of classified magnitude features

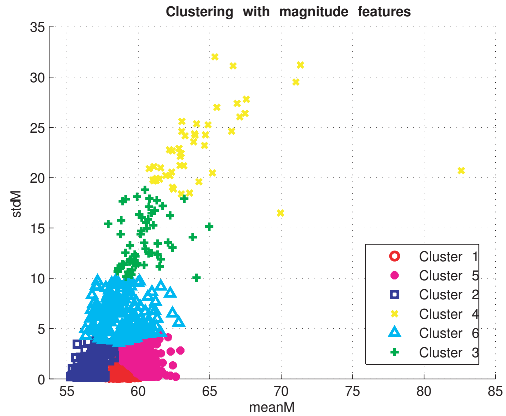

Figure 12: Map of clustered magnitude features

is roughly close to the proportion presented in Figure 10, which means the decision tree model of Tree 2 captures the most important physical motions. After carefully comparing the threshold values in Tree 2 with centroid values in Table 5, it is not hard to find the similarity between above six clusters and six defined motions {sitting, standing, walking, running, driving, bicycling}. Cluster 3, 4 and 6 have higher standard deviations than 5, so they must be intensive physical activities. According to Tree 2, cluster 3 is related to walking (or medium moving) and cluster 4 is related to running (or intensive moving). Cluster 1 occupies 80% of all data with smallest standard deviation, so it is probably related to sitting (or totally still state). Cluster 2 has a little smaller mean than that of Cluster 1, while cluster 5 has a little larger mean and standard deviation than that of both cluster 1 and 2. Inferred from Tree 2 and cluster 2's occupancy 10%, cluster 5 is related to driving (or relatively still). Cluster 2 is highly related to standing (or similar stationary activities). The difficulty lies in relating cluster 6 to bicycling. It occupies 4% of all data and cannot be bicycling related activities after we confirm with the person who collected the data. After examining the raw data, it is probably related to some transitional activities (such as from sitting to standing, from standing to walking or taking phone out of pocket etc), which has almost the same mean as sitting and standing but slightly larger standard deviation. It happens our specific bicycling behavior shares the same statistics as this cluster.

Table 5: Clustering results by k-means

Cluster No.	Centroid (meanM, stdM)	Percentage
1	(58.8689, 0.2871)	80%
2	(56.8244, 1.1331)	4%
3	(60.4005, 13.7888)	1%
4	(64.3576, 22.9281)	1%
5	(60.2872, 1.5234)	10%
6	(58.8354, 6.2127)	4%

Going back to the previous pie chart shown in Figure 10, the classified 3.3% bicycling activity is not reasonable as well. They should be some transitional activities with phone's sudden and non-periodic change in acceleration.
Based on above similarity between classified bicycling features in Figure 11 and Cluster 6 in figure 12, we change the label from bicycling to else. Since time stamps are collected together with accelerometer data, we further plot classified motion states inferred from Tree 2 along time index and generate an one week record of physical activities in Figure 13. Most of time the phone is sitting, with standing, working and driving interleaved among each other except that on Saturday, there are some running activities. If one day diary of physical activities on Monday is zoomed in Figure 14, we can see starting from 9am in the morning, the person has some activities until 8pm in the evening. There are some walking activities during the lunch time and driving activities in the morning after 10am and in the evening after 7pm. After confirmation, we have confidence the motion recognition of his daily activities is consistent with his schedule.

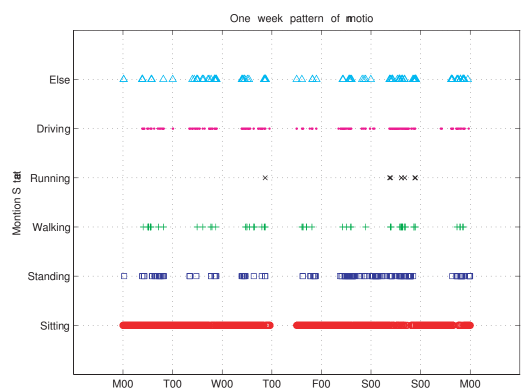

Figure 13: One week diary of physical activities

The activity diary is not smooth in transition among six activities since the static decision tree classifier must have recognition errors. We use a HMM model to smooth the output from the static decision tree classifier. The smoothing matrix (or emission matrix) defined as $E\{s_i^{'} = m | s_i = n\}$, where $m, n \in \mathcal{A}$, can be estimated according to confusion

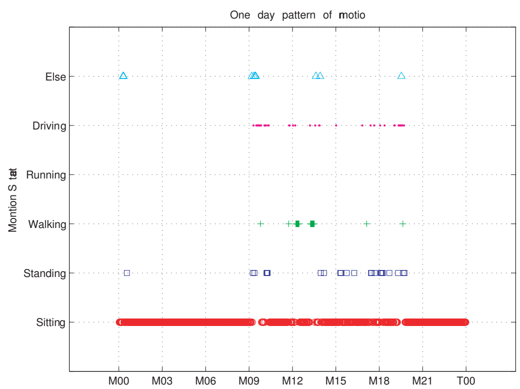

Figure 14: Monday diary of physical activities

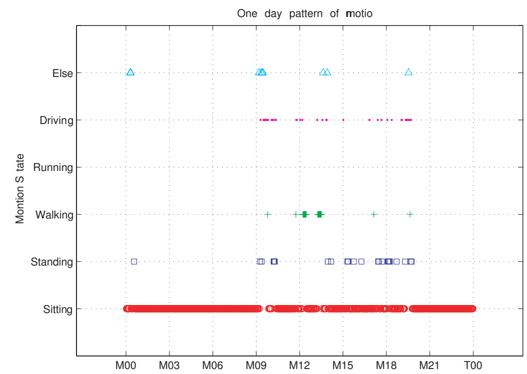

Figure 15: Monday diary of physical activities smoothed from Figure 14

matrix in Table 4.

$$E = \begin{bmatrix} 0.8406 & 0.0507 & 0 & 0 & 0.1087 & 0 \\ 0.0411 & 0.8493 & 0 & 0 & 0.1096 & 0 \\ 0 & 0 & 0.9339 & 0.0083 & 0.0413 & 0.0165 \\ 0 & 0 & 0.0625 & 0.9375 & 0 & 0 \\ 0.0545 & 0.0152 & 0.0030 & 0 & 0.9061 & 0.0212 \\ 0 & 0 & 0.0183 & 0 & 0.1376 & 0.8440 \end{bmatrix}.$$

The state transition matrix can be learned from one week classified motion sequence, which is denoted as $P\{s_{i+1} = m|s_i = n\}$, where $m, n \in \mathcal{A} = \{sitting, standing, working, running, driving, else\}$. After running an HMM model training process over 500 iterations, we obtain the following transition matrix

$$P = \begin{bmatrix} 0.9737 & 0.0043 & 0.0007 & 0.0000 & 0.0152 & 0.0061 \\ 0.0570 & 0.8764 & 0.0037 & 0.0138 & 0.0354 & 0.0137 \\ 0.1390 & 0.0516 & 0.6929 & 0.0000 & 0.0792 & 0.0372 \\ 0.0000 & 0.0733 & 0.0000 & 0.3122 & 0.3748 & 0.2397 \\ 0.0774 & 0.0314 & 0.0119 & 0.0065 & 0.7522 & 0.1207 \\ 0.1470 & 0.0231 & 0.0000 & 0.0561 & 0.5000 & 0.2738 \end{bmatrix}.$$

The transitions between six activities in above matrix look reasonable according to common sense except transition from running to other activities. The main reason is there are few running activities available in the one week motion sequence. Using above E and P in Viterbi algorithm, it corrects 6.25% out of the original sequences. A smoothed one day diary of physical activities is shown in Figure 15. Comparing the results in Figure 15 with those in Figure 14, there are more observable walking activities around lunch time and driving activities before and after work. However, the abnormal driving activities during the work time are smoothed out and eliminated by half. The smoothing process shows the motion recognition accuracy of a static decision tree classifier can be improved by a second-level temporal HMM classifier.

7. CONCLUSION AND FUTURE WORK

In this paper, we have investigated physical motion recognition using mobile phones with built-in accelerometer sensors. We take a different approach from the prior art that involves rigid training data collection under supervision with wearable external sensors. Data processing and smoothing techniques are discussed first to reduce the noise present in accelerometer data and improve its quality. Since phone's position on a human body is varying from person to person, its orientation cannot be partially fixed or relatively fixed. We explore orientation-independent features extracted from magnitudes as well as vertical and horizonal components in acceleration for six common physical activities, such as sitting, standing, walking, running, driving etc. We find decision tree achieves the best performance among four different static classifiers, while vertical/horizonal features have better performance than magnitude features. Furthermore, a well-pruned decision tree with simple time-domain features and less over-fitting on the training data can provide a usable model for generating an activity diary. After applying one decision tree model to a large set of one week's unlabeled accelerometer data and comparing the results with those from k-means clustering method, we find a significant similarity between each other. Finally we discuss how to refine labels based on feature observation and hidden ground truth from raw data. The generated activity sequence with time stamps classified by proposed static decision tree model can be smoothed by an HMM-based Viterbi algorithm to maximize the utilization of temporal correlation.

One potential application for physical activity recognition is in the domain of mobile healthcare, like activity-aware computing introduced in [16] that enables users to move from activity-based interaction toward activity-aware engagement. The other possible application is human-centric sensing and sharing in mobile social networks, such as microblogging service discussed in [6] that mobile devices to record content from sensors and share this content in a real-time.

Although this paper has investigated several important aspects of physical activity recognition using mobile phones, such as phone-orientation independent features, low-weight time-domain feature calculation for mobile CPU and general static classifiers for common human motions, there are still lots of open research problems worth further consideration.

- Sensor data calibration. We notice the accelerometer data collected from different phones has different gravity measurements, as shown in Figure 4. Even for the same phone, one stationary orientation (facing up) has distinct magnitude readings of acceleration from the other (facing down). How to calibrate accelerometer sensor readings and normalize them to the same scale across mobile phones is an interesting problem.

- Model adaption. In our simplified decision tree models, the mean is used to classify sitting and standing based on uncalibrated sensor readings. The mean variation depends on accelerometer sensors in the phone, because sometimes smaller mean is standing while sometimes smaller mean is sitting. So the threshold should be adaptive. Person's driving behavior and road condition have certain impact on classification since smooth driving is very close to sitting. How to generate an adaptive model accommodating each case is a necessary step to make the model practical. Introducing more features from other sensors like GPS signal may help solve this problem.

- Dynamic classifier. Static classifiers like decision trees are used through the paper. HMM model is only used for smoothing the classified activity sequence. If HMM model or similar Markov model is built directly on a sequence of extracted features, the overall classification accuracy can be improved since more temporal correlation is captured in the model.

8. ACKNOWLEDGEMENT

The author would like to thank his colleagues Zhigang Liu, Guang Yang and Hawk-Yin Pang for collecting experimental accelerometer data. He would give special thanks to the C3 team at Nokia Research Center for their excellent data collection tool, "Nokia Simple Context". He also wants to thank Zhigang Liu for his robust Python script "Accelerometer Sampler" for instant data collection and Hong Lu for his constructive comments toward some parts of the paper.

9. REFERENCES

[1] Nokia simple context. https://simplecontext.com/eb2/.

[2] Nokia sports tracker. https://sportstracker.nokia.com/nts/main/index.do.

[3] Nokia step counter. http://betalabs.nokia.com/betas/view/nokia-step-counter.

[4] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive*, pages 1–17, 2004.

[5] Y. EL-Manzalawy and V. Honavar. *WLSVM: Integrating LibSVM into Weka Environment*, 2005. Software available at http://www.cs.iastate.edu/ yasser/wlsvm.

[6] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 174–186, New York, NY, USA, 2008. ACM.

[7] Y. Kawahara, H. Kurasawa, and H. Morikawa. Recognizing user context using mobile handsets with acceleration sensors. In *PORTABLE '07: IEEE International Conference on Portable Information Devices*, pages 1–5, Orlando, FL, 2007.

[8] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Troster. Wearable sensing to annotate meeting recordings. In *Proceedings of the Sixth International Symposium on Wearable Computers ISWC*, pages 186–193, 2002.

[9] K. Kunze and P. Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 20–29, New York, NY, USA, 2008.

[10] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster. Where am i: Recognizing on-body positions of wearable sensors. In *In: LOCAŠ04: International Workshop on Locationand Context-Awareness*, pages 264–275. Springer-Verlag, 2005.

[11] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. *Lecture Notes in Computer Science : Pervasive Computing*, pages 1–16, 2006.

[12] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 766–772, 2005.

[13] 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 *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, New York, NY, USA, 2008.

[14] D. Mizell. Using gravity to estimate accelerometer orientation. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 252, Washington, DC, USA, 2003. IEEE Computer Society.

[15] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. *IAAI-05: American Association for Artificial Intelligence*, 2005.

[16] M. Tentori and J. Favela. Activity-aware computing for healthcare. *Pervasive Computing, IEEE*, 7(2):51–57, 2008.