Real-time Activity Recognition using Smartphone Accelerometer

Shuang Na
Department of Mathematics & Statistics
University of South Florida
Email: sna@mail.usf.edu

Kandethody M. Ramachandran
Department of Mathematics & Statistics
University of South Florida
4202 E Fowler Ave, CMC 342, Tampa, Florida,USA
Email: ram@usf.edu

Ming Ji
College Of Nursing
University of South Florida
Email:mji@health.usf.edu

Yicheng Tu
Department of Computer Science and Engineering
University of South Florida
Email:tuy@mail.usf.edu

***Abstract*—To identify the real-time activities, an online algorithm need be considered. In this paper, we will first segment entire one activity as one-time interval using Bayesian online detection method instead of fixed and small length time interval. Then, we introduce two-layer random forest classification for real-time activity recognition on the smartphone by embedded accelerometers. We evaluate the performance of our method based on six activities: walking, upstairs, downstairs, sitting, standing, and laying on 30 volunteers. For the data considered, we get 92.4% overall accuracy based on six activities and 100% overall accuracy only based on dynamic activity and static activity.**

***Keywords*—Bayesian online detection; Human activity recognition; Random forest**

I. Introduction

It has been estimated that there would be 6.1 billion smartphone users by 2020, which is about 70% of the population worldwide [1]. The smartphone is not only a communication equipment but also a powerful tool with a variety of functions, such as photography, radio, calculator, game, and GPS, which contains camera, accelerometer, microphone, magnetic compasses, and GPS sensors, etc. The human activity recognition (HAR) problem using a smartphone with built-in sensors include accelerometer and GPS that allows continuous monitoring human activity patterns (i.e., walking, running, sitting, etc.) of the user who carries a smartphone every day as well as personal activity tracking. For instance, some people are concerned about how many calories they burn every day through exercise, or detecting whether an elder fell and whether he/she require emergency aid. By comparing with other wearable sensors, it is more convenient and reasonable to carry the cell phone every day due to the small size and multi-functionality of the smartphone.

The first project of HAR is discussed in [2] in the late '90s. During 25 years, there were several approaches to improve the identifying process by wearing multiple sensors that placed in several locations of the body ([3],[4],[5]). In [3], which indicates that multiple accelerometers distributed on the different areas of a body have more effective recognition ability. In addition, the application of complex methods such as statistics learning and machine learning also effect on improving the identifying process ([6], [7], [8], [9]). An overview of HAR problem by carrying wearable sensors is discussed in [10], which compares HAR system design, sensors selection, recognition method and evaluation systems. It claims that it is a challenge to define the best detecting procedure because of considering different tasks, sampling rates, algorithms, computing speed and evaluation methods. Nevertheless, more sensors give better results in general. However, it is unreasonable to wear multiple types of equipments every day because of expense, complexity, and inconvenience. Smartphone as a much used electronic product that has been widely applied for activity recognition (e.g. [7],[11],[12],[13], [14]). Therefore, the research field of smartphone based HAR becomes very important.

We present a real-time human activity recognition algorithm using sensor data generated from the triaxial accelerometer built-in smartphone [13]. The six daily life activities discussed here are: walking, walking upstairs, walking downstairs, sitting, standing and laying. To deal with HAR problem, first of all, we must partition the entire time interval into segments because the raw data cannot be used in classification algorithm directly. Then, extract meaningful features (mean, standard deviation, peak, etc.) from every segment, which are applied to the classification algorithm, such as decision tree. Finally, we will train and test the identifying process. Because of limitation of cell phone's battery and CPU, we cannot train the model on a cell phone. Usually, the training process is either already learned on a computer or happening on cloud computing through the Internet. However, technology is developing with time, more and more types of cell phones are allowed to run training algorithm on it [14]. Therefore, the user can collect data and identify activities on their phone; it is more convenient when a user wants to adapt the activity training model or when the Internet is not available. On the other hand, it is also desirable to develop more efficient and less complex learning models so that training procedure can be processed on the cell phone. For the classification method, there are two types of algorithm: online and offline learning ([15],[16]). By comparing with offline learning method, online learning can be applied to real-time training data because it is able to adapt the model as new data being collected.

In this paper, we introduce a combination of Bayesian online detection algorithm and two-layer random forest classification, which can automatically identify human activities at a certain time without user intervening, such that user does not need to mark start time and end time for every activity. Using Bayesian online segmentation method to detect the time interval, we extract meaningful features based on a entire activity time interval that considered as a segment. These features from a entire activity contain more information than short time interval generated by fixed length time segmentation. For the training procedure, different classification layers use different features (section 2). The first layer aims to distinguish dynamic activities (walking, walking upstairs and walking downstairs) versus static activities (sitting, standing and lying). The main feature used in the first layer classifier is amplitude defined as $\sqrt{x_{i,1}^2 + x_{i,2}^2 + ... + x_{i,p}^2}$, x_i is i^{th} point with p-dimension. During the second layer classification processing, there are two separated classification training models A and B to classify different sub-classes. If we get static activity on the first layer for a certain interval, then we go through model A at second layer to further classify as sitting, standing or lying; otherwise, we go through model B at second layer to further partition into walking, walking upstairs or walking downstairs.

This paper is organized as follows. Section II presents the Bayesian online segmentation. Section III discusses two-layer classification method. In section IV, we present the result of applying the methods introduced to a real-time activity data. The conclusion is given in section V.

II. Methodology

For these developed human activity recognition (HAR) methods, the pre-processing procedure of HAR methods is to separate the whole time series interval into small segments. The reason for this step is important and necessary is that the raw individual observation cannot be applied directly in classification methods. In most of the literature, one partition the whole time series into equal length segments, the length of the segment could be 1.5s, 2s, and 3s, etc. However, these type of intervals are so short that cannot capture enough information of activities. In this paper, we complete this step based on Bayesian online segmentation method introduced in section 2.1. This method detects change points between two patterns, and it defines the time series between two change points as a segment. This procedure results in varying length of time intervals rather than equal length of time interval and small segments. Next step, we need to extract features from each segment. The feature widely used so far are categorized as time-domain features, frequency-domain features, wavelet features and Heuristic Features [17]. Here, we mainly use time-domain features and frequency-domain features. The review paper [10] discusses current classification methods that include decision tree, k-nearest neighbors, Bayesian, neural network, fuzzy logic, support vector machines, classifier ensembles, regression methods and Markov models that are applied widely in many publications and applications. In this work, we introduce two-layer random forest method for activity classification, we use accuracy and confusion matrix as the evaluation index. Fig 1 displays the process of real-time activity recognition.

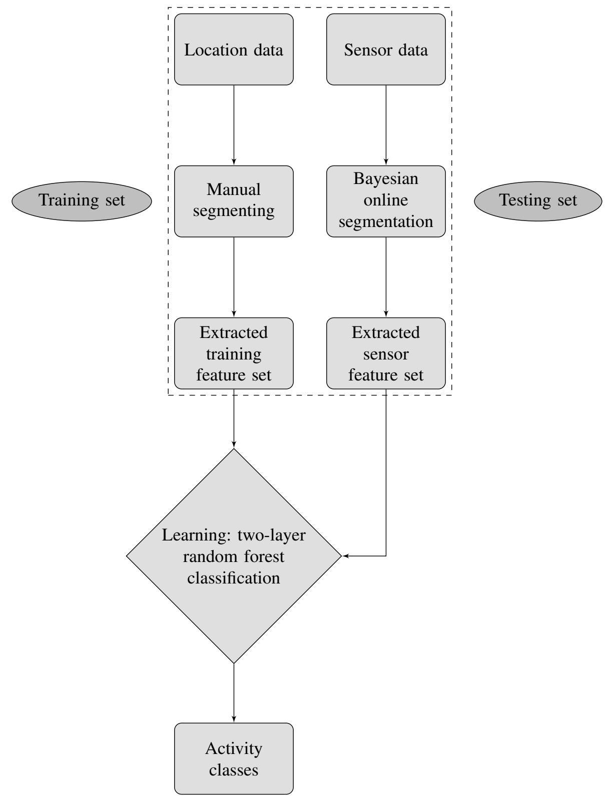

Fig. 1. Online Activity Recognition Process

A. Bayesian Online Segmentation

In this paper, we consider the three-dimensional data points $x_t = (x_{t1}, x_{t2}, x_{t3})$ at time t with frequency 50Hz that is generated by smartphone accelerometer. The task of classifying the human patterns from these sensor data is to group homogeneous data sets and separate heterogeneous data sets. These observations are listed on timeline, all observations between two change points construct a time series segment defined as a pattern. Those homogeneous observations are assumed to follow a multi-normal distribution, and different patterns follow different multi-normal distributions. Therefore, to find the change point between two patterns becomes a significant problem.

In [18] and [10], an overview of the human activity recognition process and discussion of segmenting methods. Bayesian online detection method [19] can be used to prepare these segments automatically for classification. First, we consider the concept of "run length" r_t, which is the length of the current run at time t and it is linear about time t. For example, if $r_t = 0$ at t=8, x_8 is a change point; if $r_t \neq 0$, we keep run one more and repeat the process. $x_t^{(r)}$ is defined as the set correspond to run length r_t. If r_t is zero, $x^{(r)}$ is an empty set. For example, t=9, $r_t = 1$, then $x_9^{(r)} = \{x_8, x_9\}$. To find the posterior distribution $P(r_t|x_{1:t})$, we need to generate a recursive joint distribution $P(r_t, x_{1:t})$,

$$\begin{aligned} P(r_t|x_{1:t}) &= \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \propto P(r_t, x_{1:t}) \\ &\propto \sum_{r_{t-1}} P(r_t, r_{t-1}, x_t, x_{1:t-1}) \\ &\propto \sum_{r_{t-1}} P(r_t, x_t|r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &\propto \sum_{r_{t-1}} P(r_t|r_{t-1}) P(x_t|r_{t-1}, x_{t-1}^{(r)}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{1}$$

Here, $P(r_t|r_{t-1})$ is a prior probability, the joint distribution $P(r_t, x_{1:t})$ is called growth probability and $P(x_t|r_{t-1}, x_{t-1}^{(r)})$ is a predictive probability. At every time recursion, we pick the r_t with the largest posterior probability, which r_t is also associated with the largest joint distribution in recent data. Next, the prior distribution $P(r_t|r_{t-1})$ and the predictive distribution $P(x_t|r_{t-1}, x_t^{(r)})$ is defined in following steps.

The run length has two directions: one direction is that no change point happens at time t and $r_t = r_{t-1} + 1$, which means the new data still stays in the same group and follows the same distribution; another direction is that a change point occurs, r_t drop to 0 with probability $H(r_t) = 1/\lambda$. Here, $H(r_t)$ is hazard function based on geometric distribution with parameter λ [20]. The prior distribution is:

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}) & \text{if } r_t = r_{t-1} + 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The predictive probability $P(x_{t+1}|r_t, x_t^{(r)})$ is a marginal distribution that integral the parameter vector θ correspond to the current run length r_t. The r_t depends on the recent data set $x_t^{(r)}$, which set keep the homogeneous observations and the generated distribution won't change with time. Define the predictive probability as follows:

$$P(x_{t+1}|r_t, x_t^{(r)}) = \int P(x_{t+1}|\theta) P(\theta_t^{(r)} = \theta|r_t, x_t^{(r)}) d\theta \tag{3}$$

Here, $\theta_t^{(r)}$ is the parameter of current run length. Assume the tri-dimensions sensor data $x_t = (x_{t1}, x_{t2}, x_{t3})^T$ follow three multiple normal distributions with mean μ and inverse-covariance matrix $\Omega = \Sigma^{-1}$ and dimension $d = 3$. The likelihood function of n data points is described as follows:

$$P(X_{1:t}|\mu, \Omega) \tag{4}$$

$$= (2\pi)^{-nd/2} |\Omega|^{t/2} exp\Big(- \frac{1}{2} \sum_{i=1}^{t} (x_i - \mu)^T \Omega (x_i - \mu) \Big)$$

$$\propto |\Omega|^{t/2} exp\Big(- \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)^T \Omega (x_i - \mu) \Big) \tag{5}$$

For the prior distribution $P(\mu, \Omega)$, assume $\mu \sim \mathcal{N}(\mu_0, (\kappa_0 \Omega)^{-1})$ normal distribution and $\Omega \sim Wi_d(T_0, \nu_0)$ Wishart distribution, so the prior distribution of combining both unknown parameters are:

$$P(\mu, \Omega|\mu_0, \kappa_0, \nu_0, T_0)$$

$$\propto |\Omega|^{1/2} exp\Big(- \frac{1}{2} (\mu - \mu_0)^T (\kappa \Omega)(\mu - \mu_0) \Big) |\Omega|^{(\nu_0 - d - 1)/2} \tag{6}$$

$$\times exp\Big(- tr(T\Omega)/2 \Big) \tag{7}$$

Hence, the posterior distribution is:

$$P(\mu, \Omega|X_{1:t}) \tag{8}$$

$$\propto P(X_{1:t}|\mu, \Omega) P(\mu, \Omega) \tag{9}$$

$$\propto \mathcal{N}(\mu|\mu_t, \kappa_t) Wi_d(\Omega|\nu_t, T_t) \tag{10}$$

where,

$$\begin{aligned} \kappa_t &= \kappa_0 + t \\ \mu_t &= \frac{\kappa_0 \mu_0 + t\bar{X}_t}{\kappa_0 + t} \\ \nu_t &= \nu_0 + t \\ T_t &= T_0 + \sum (X_i - \bar{X}_t)(X_i - \bar{X}_t)^T \end{aligned} \tag{11}$$

$$+ \frac{\kappa_0 t}{\kappa_0 + t} (\mu_0 - \bar{X}_t)(\mu_0 - \bar{X}_t)^T \tag{12}$$

Therefore, the update step for every one new data.

$$\begin{aligned} \kappa_{t+1} &= \kappa_t + 1 \\ \mu_{t+1} &= \frac{\kappa_t \mu_t + X_{t+1}}{\kappa_t + 1} \\ \nu_{t+1} &= \nu_t + 1 \\ T_{t+1} &= \frac{\kappa_t (X_{t+1} - \mu_t)(X_{t+1} - \mu_t)^T}{\kappa_t + 1} \end{aligned} \tag{13}$$

Finally, the posterior predictive probability is

$$P(X_{t+1}|X_t^r, r_t) = t_{\nu_t - d + 1}(\mu_t, \frac{T_t(\kappa_t + 1)}{\kappa_t(\nu_t - d + 1)}) \tag{14}$$

When new data comes, the algorithm updates parameter and the joint distribution $P(r_t, x_{1:t})$, which approximates to posterior distribution $P(r_t|x_{1:t})$. This Bayesian method create t posterior distribution $\{P(r_t|x_{i:t})\}\}_{i=1}^{t}$ at every iteration time t, to pick up the r_t with highest posterior probability. If r_t change to 0 that means a new segment, x_t is defined as a change point. More detail is shown in Appendix.

Adjust Bayesian Online Method

Time consumption is one of the important issues of the Bayesian online algorithm because the computing time is linear with the number of observations, which makes computing time being linearly increasing with time. Here, we introduce sliding window method that is used to construct a fixed window size N, which is so large that includes at least a few activity segments. In this experiment we take $N = 10000$, that's equivalent to 200s. Depending on different data set, we could choose larger N especially in cases where a single activity takes more time. The adjusted Bayesian online method is to re-initial all parameters when starting a new sliding window and eliminates all old parameter information of last sliding window. The algorithm of adjusted Bayesian online segmenting shows in Fig 2.

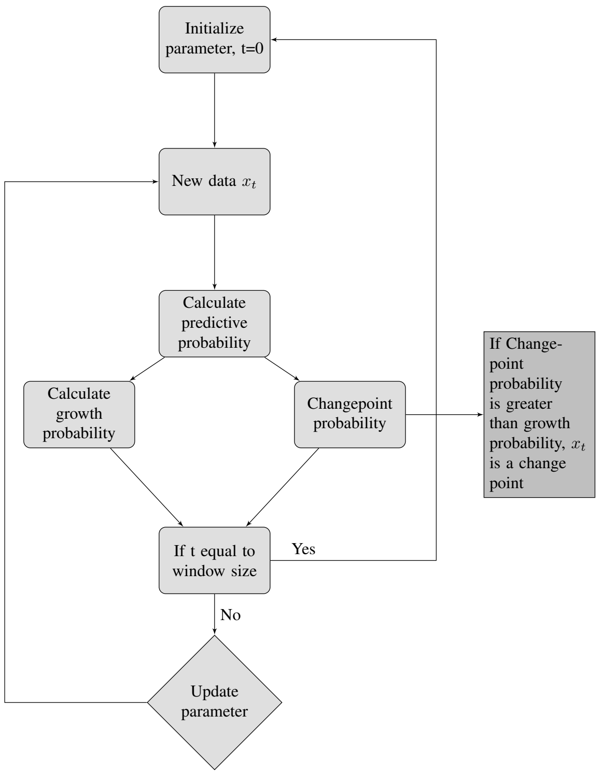

Fig. 2. Bayesian Online Segmentation

B. two-layer Classification

In this experiment, instead of extracting features from every small segment, such as 2s, we consider extracting features from the entire activity time segment constructed automatically by Bayesian Online Detection method. One reason for considering the feature of a whole segment is that less feature calculation than feature computation of every 2s. Another reason is to classify the whole sequence rather than classifying subsequence that subsequence contains less information of activity process. In addition to the three-dimensional data (x_i, y_i, z_i), there is another important measurement: the distance of a observation to the original point $(0, 0, 0)$ called magnitude, which is defined as $d_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$. The feature we used here are following six type criterion:

1. Average of (x_i, y_i, z_i, d_i);
2. Standard deviation of (x_i, y_i, z_i, d_i);
3. Average of local maximum of (x_i, y_i, z_i, d_i);
4. Average of time difference between two consecutive local maximums of (x_i, y_i, z_i, d_i);
5. Skewness of (x_i, y_i, z_i);
6. Kurtosis of (x_i, y_i, z_i).

The observation can approximate as a periodic wave, and the repetitive peaks can regard as one of the characteristics that are used to distinguish different axes of different activities [12]. Because the waves of dynamic activities repeat quicker than static activities, the time gap between two consecutive peaks of dynamic activities is shorter than static activities'. The time gap between two consecutive peaks is also used as the indexes to distinguish the activities.

After segmenting time series, this algorithm trains two-layer classification. The first layer classifies the activities into two categories: dynamic activities (walking, walking upstairs, walking downstairs) and static activity (standing, sitting, lying). The training features of first layer classification only include the first four criterion of d_i: average, standard deviation, average of local maximum and average of time difference between two consecutive local maximums. These criterion of d_i are the most essential factors to distinguish dynamic and static patterns. For the second layer classification, there are two separated classification processes. One classifier trains and tests only on dynamic activity results into three classes: walking, upstairs and downstairs. Another classifier trains and tests on static activity that also result in three classes: sitting, standing and lying. Here, we apply random forest as the classification method, which is the most widely used classifier and suitable for many types of data [14].

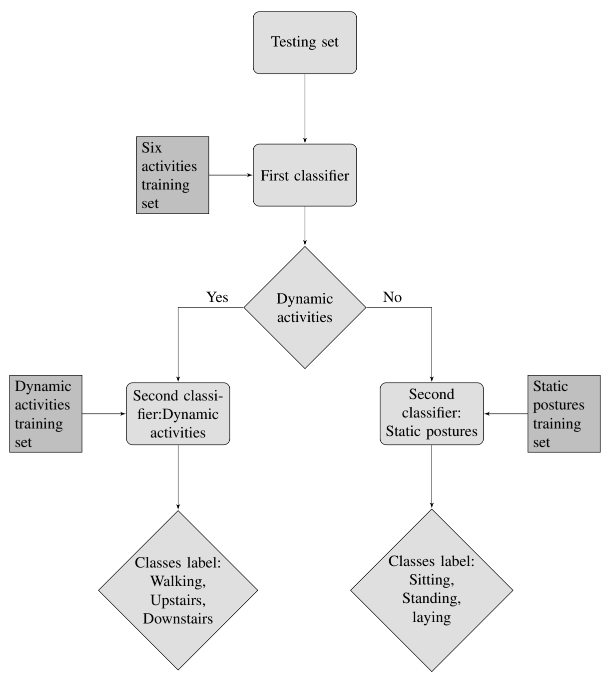

Fig. 3. two-layer Classification Flow Chat

The two-layer classification algorithm 1is displayed in Fig 3.

Algorithm 1 two-layer Classification algorithm

1: **procedure** GENERATED NEW SEGMENT
2: Create features from generated new segment $x_t^{(r)}$
3: Predict the activity of new segment set using the first layer features
4: If the predictive label from step 3 is dynamic activities, go to step 5; if the predictive label is static activities, go to step 6;
5: Predict the activity of new segment set using second layer features considering dynamic activities
6: Predict the activity of new segment set using second layer features considering static activities
7: **end procedure**

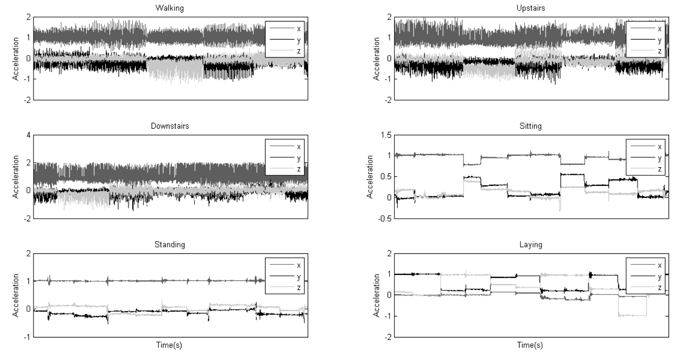

Fig. 4. Acceleration plot of six activities

III. EXPERIMENT

The data source is provided by [21], where the experiments have been carried out with a group of 30 volunteers. They performed a protocol of activities composed of six daily activities: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs). All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution. The captured 3-axial linear acceleration at a constant rate of 50Hz using the embedded accelerometer. Randomly choose 200s observations of a few volunteers to show the tri-axis observation of six activities in Fig 4. It is relatively easy to distinguish dynamic activities and static activities because the dynamic activities have stronger fluctuation than static activities. However, it is a challenge to identify the walking, walking upstairs and walking downstairs in the dynamic group, as well as identifying sitting, standing and laying in the static group. Also, volunteers generate their particular wave stream that observations are entirely different with each other when they even perform the same action. For instance, in Fig4_testing, the z-axis data of around first 80s is less fluctuating than the z-axis data from 80s to 120s. On the other hand, the observations of different activities exhibited from different persons that could be same. Therefore, the combination of the sensor data from all volunteers is difficult to be detected due to the differences in the pattern. The reason of training on combining data is that prototype created by such training set generally can be applied so that the user does not require to record the start and stop time for specific activities.

The obtained dataset was randomly partitioned into two sets, where 70% of the data was selected as training data and 30% as the test data. The training set is applied to train two-layer random forest to get the prototypes model. Subsequently, the model is used to classify the online segments using Bayesian online segmentation discussed before and come with a class label as output. First of all, let's check the automatically segmenting confusion matrix result in Table I using Bayesian online segmentation algorithm, those errors are caused by segmenting bias and delay with real boundaries. Here, we use the majority vote to label the class of each segment.

TABLE I
BAYESIAN ONLINE SEGMENTATION CONFUSION MATRIX

	Walking	Walking Upstairs	Walking Downstairs	Sitting	Standing	lying
Walking	1	0	0	0	0	0
Walking Upstairs	0	0.9601	0.0399	0	0	0
Walking Downstairs	0	0.0451	0.9549	0	0	0
Sitting	0	0	0	0.9533	0.0265	0.0087
Standing	0	0	0	0.0477	0.9415	0
Laying	0	0	0	0.0078	0.0058	0.9749

In Table I, the average error rate of bias and delay is 3.55%, which shows this algorithm can automatically and efficiently detect changing and find activities time interval. To compare with Bayesian online segmentation method, we use Sliding Window and Bottom-up(SWAB) [22] online segmentation algorithm as the optional choice. The SWAB is to process bottom-up algorithm during a large enough sliding window that can include a few segments. Unlike the Bayesian method, SWAB requires defining initial minimal segmentation length and final merged number of segments. In TableII, we display overall accuracy based on the different scale of minimum length and number of segments. The accuracy is increasing as long as we increase the number of segments and increase the minimum length. However, increasing both criteria will result in segmentation meaningless. Therefore, we are uncertain to choose the best option. Compared with that, the Bayesian method has less pre-defined requirements and hence more desirable.

Table III shows the testing accuracy of two times classifier based on Bayesian online segmentation. The overall accuracy

TABLE II
SWAB OVERALL ACCURACY

	min len = 2	min len = 5	min len = 10	min len = 20	min len = 30
num seg= 25	0.8417	0.8444	0.8494	0.8793	0.8991
num seg= 50	0.8854	0.8914	0.9022	0.9329	0.9534
num seg= 100	0.9227	0.9386	0.9514	0.9688	0.9931
num seg= 200	0.9658	0.9663	0.9739	0.9992	0.9981

is 92.4% for six states, the accuracy of static postures and dynamic activities are 100 % in this experiment. By the overall accuracy of two groups, we can 100% detect whether the person moves or not. From the estimated change points, we can estimate the length of time for a person to be active or sedentary.

TABLE III
ONLINE TWO-LAYER CLASSIFICATION CONFUSION MATRIX

	Walking	Walking Upstairs	Walking Downstairs	Sitting	Standing	lying
Walking	26	3	0	0	0	0
Walking Upstairs	3	50	1	0	0	0
Walking Downstairs	0	3	51	0	0	0
Sitting	0	0	0	34	2	0
Standing	0	0	0	5	30	1
Laying	0	0	0	0	0	36

For showing the reasoning of choosing random forest, we compared the results of training by KNN, and SVM, Boosting with Random forest in Fig5. The overall accuracy of Boosting and Random forest is very close and both higher than other two algorithms. Checking the accuracy of every activity, Boosting and Random forest generated result is better than others except for walking upstairs that SVM performs the best. Even the complexity of KNN $O(nk + nd)$ (k is a pre-defined number of classes) shows that KNN is an efficient classification algorithm, but KNN provides poor identify result. Considering SVM has time complexity $O(dn^2)$ that SVM is more effective in cases where the number of dimensions is greater than the number of samples. Taken as a whole, Boosting and Random forest is the optimal choices. Nonetheless, the time complexity of boosting is $O(ndKlog(n))$ that is higher than the time complexity of random forest $O(ndlog(n))$. Here, n is the number of observation, d is the number of features, and K is depth. Consider the task of real-time pattern detecting, not only accuracy is important, but less time consuming is also another criterion. Therefore, the random forest is the better way to handle our data.

Based on Table I segmenting outcomes, we can simultaneously input the result into two times classifier and output activity label immediately.

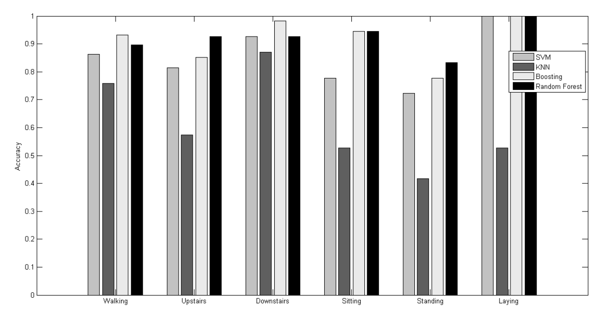

Fig. 5. Accuracy of six activities using four classification methods (Bayesian segmentation)

A. Comparing OBS with SWAB

This data source is provided by Dr. Tu Yicheng's Lab at University of South Florida, which is generated by iPhone 6 built-in accelerometer with 50Hz. The sensor data is tri-dimensional: x-axis shows the acceleration rate on forth-back direction; y-axis shows the acceleration rate on left-right direction; z-axis shows the acceleration rate on up-down direction. The sensor data include sitting, standing, walking_upstairs, walking_downstairs, walking and jogging six activities. However, the label of each activity is unknown. Comparing the change point detecting result by OBS method in Figure 6 with SWAB state estimation result, the OBS algorithm does an excellent performance.

SWAB method is considered as the alternative segmentation method comparing with OBS. The length of sliding window is 1000, which is long enough to contain a few activities. Since these data set doesn't include actual activity labels and it needs visually check whether or not the segmentations are appropriate. SWAB requires the minimum length of starting segments and the fixed number of segments per sliding

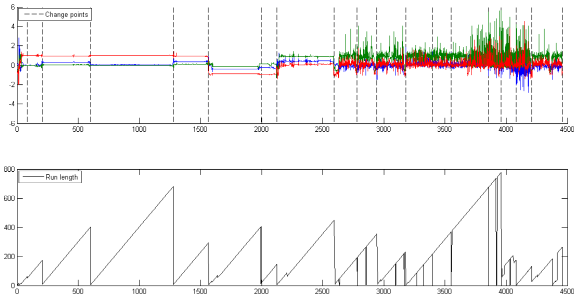

Fig. 6. Acceleration plot of six activities

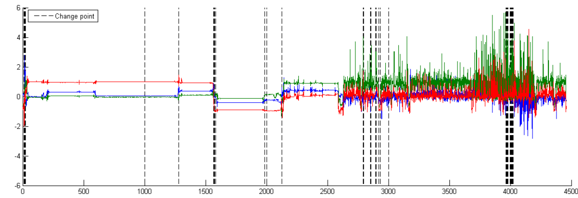

Fig. 10. SWAB with min length=2, number of segments=10

window. Based on the result displayed in Figure 7,8,9 and 10, the change point detecting is sensitive to the fluctuating observations (observation 2500th to 4500th). Overall, OBS can be considered as the better choice for activity detecting issue.

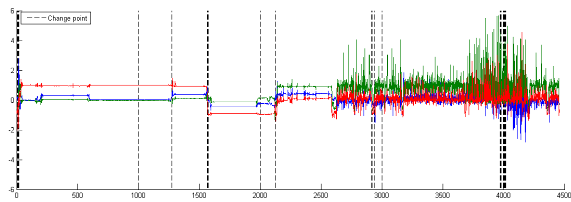

Fig. 7. SWAB with min length=5, number of segments=5

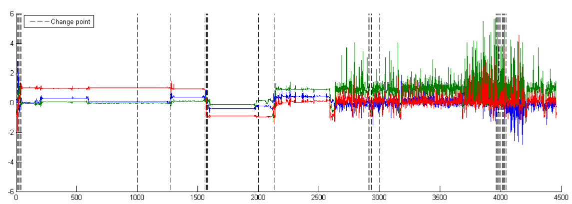

Fig. 8. SWAB with min length=10, number of segments=5

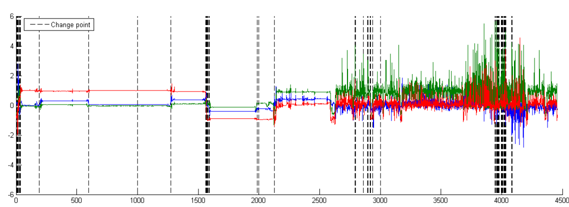

Fig. 9. SWAB with min length=5, number of segments=10

IV. Conclusion

We present a real-time activity recognition algorithm, which is used to apply on smartphone platform for human activity identification. We introduce a Bayesian online detection algorithm and two-layer random forest classifier. Rather than extracting features from small segments that one activity time interval might contain a few such segments, this method computes features based on entire activity time interval. Different classify layers use different features to avoid meaningless features and save running time. The first layer aims to distinguish dynamic activities (walking, upstairs and downstairs) and static activities (sitting, standing and lying) using the amplitude as the key feature. The process of the second layer is to distinguish the three behaviors in different categories. However, it is still a big challenge to differentiate walking and walking upstairs very well only based on accelerometer sensor data. We could get a better result with regarding GPS data in the later work. In this work, performance has reached with overall accuracy for six states is 92.4%, and the overall accuracy for two categories: dynamic activities and static activities is 100%. For USF lab data, we cannot check the segmentation results between OBS and SWAB by quantitative measurement since the activity label is unknown. By comparing the detecting boundary result with observations, the OBS give better segmenting result that can detect change point appropriately even dynamic activities. Further, OBS method doesn't need prior information as well as SWAB method such as the number of segments. In the future, a real-time detection activity system can be designed on a smartphone. Since it has been trained and it can automatically detect behavior change, therefore it's easy to handle without user knowledge of machine learning and instruct smartphone when user change behaviors. Even though the multivariate normality assumption is bit restrictive, we get good results. We plan to relax this assumption in the future work. Further, we also can extend the set of activity with low frequency and high frequency, such as quick walking, slow walking, fast running, and jogging, etc.

Appendix

The prior distribution of combination of normal distribution and Wishart distribution is:

$$
\begin{aligned}
&P(\mu,\Omega|\mu_0,\kappa_0,\nu_0,T_0)\\
&=\mathcal{N}(\mu|\mu_0,\kappa_0)Wi_d(\Omega|\nu_0,T_0)\\
&=(2\pi)^{-d/2}|\kappa_0\Omega|^{1/2}exp\Big(-\frac{1}{2}(\mu-\mu_0)^T(\kappa\Omega)(\mu-\mu_0)\Big)|\\
&\times\Omega|^{(\nu_0-d-1)/2}exp\Big(-tr(T\Omega)/2\Big)2^{-\nu_0 d/2}|T|^{-\nu_0/2}\Gamma_d(\nu_0/2)
\end{aligned}
$$

$$\propto |\Omega|^{1/2} exp\Big(-\frac{1}{2}(\mu-\mu_0)^T(\kappa\Omega)(\mu-\mu_0)\Big)|\Omega|^{(\nu_0-d-1)/2}$$
$$\times exp\Big(-tr(T\Omega)/2\Big) \tag{15}$$

Therefore, we can get the posterior distribution:

$$P(\mu,\Omega|X_{1:t}) = \frac{P(\mu,\Omega,X_{1:t})}{P(X_{1:t})} = \frac{P(X_{1:t}|\mu,\Omega)P(\mu,\Omega)}{P(X_{1:t})}$$
$$\propto P(X_{1:t}|\mu,\Omega)P(\mu,\Omega)$$
$$\propto |\Omega|^{t/2} exp\Big(-\frac{1}{2}\sum_{i=1}^{t}(x_i-\mu)^T\Omega(x_i-\mu)\Big)$$
$$\times |\Omega|^{1/2} exp\Big(-\frac{1}{2}(\mu-\mu_0)^T(\kappa\Omega)(\mu-\mu_0)\Big)|\Omega|^{(\nu_0-d-1)/2}$$
$$\times exp\Big(-tr(T\Omega)/2\Big)$$
$$\propto |\Omega|^{1/2}|\Omega|^{(\nu_0+t-d-1)/2}$$
$$\times exp\Big(-\frac{1}{2}(\mu-\frac{\kappa_0\nu_0+t\bar{X}}{\kappa_0+t})^T(\kappa\Omega)(\mu-\frac{\kappa_0\nu_0+t\bar{X}}{\kappa_0+t})\Big)$$
$$\times |\Omega|^{(\nu_0-d-1)/2} exp\Big(-tr(T\Omega)/2\Big)$$
$$\times exp\Big((T_0+\sum(X_i-\bar{X})(X_i-\bar{X})^T$$
$$+\frac{\kappa_0 t}{\kappa_0+t}(\mu_0-\bar{X})(\mu_0-\bar{X})^T)\Big)$$
$$\propto |\Omega|^{1/2}|exp(-\frac{1}{2}(\mu-\mu_t)(\kappa_t\Omega)(\mu-\mu_t)^T)$$
$$|\Omega|^{(\nu_t-d-1)/2} exp\Big(T_t\Omega\Big)$$
$$\propto \mathcal{N}(\mu|\mu_t,\kappa_t)Wi_d(\Omega|\nu_t,T_t) \tag{16}$$

For the update step,

$$\mu_{t+1}-\mu_t$$
$$= \frac{\kappa\mu_0+(t+1)\bar{X}_{t+1}}{\kappa+t+1} - \frac{\kappa\mu_0+t\bar{X}_t}{\kappa+t}$$
$$= \frac{\kappa X_{t+1}+tX_{t+1}-\kappa\mu_0-t\bar{X}_t}{(\kappa+t+1)(\kappa+t)} = \frac{X_{t+1}-\mu_t}{\kappa_t+1}$$
$$\mu_{t+1} = \mu_t + \frac{X_{t+1}-\mu_t}{\kappa_t+1} = \frac{\kappa_t\mu_t+X_{t+1}}{\kappa_t+1} \tag{17}$$

$$T_{t+1}-T_t$$
$$= X_{t+1}X_{t+1}^T-(t+1)\bar{X}_{t+1}\bar{X}_{t+1}^T+t\bar{X}_t\bar{X}_t^T$$
$$+\frac{\kappa(t+1)}{\kappa+t+1}(\mu_0\mu_0^T-2\mu_0\bar{X}_{t+1}^T+\bar{X}_{t+1}\bar{X}_{t+1}^T)$$
$$-\frac{\kappa t}{\kappa+t}(\mu_0\mu_0^T-2\mu_0\bar{X}_t^T+\bar{X}_t\bar{X}_t^T)$$
$$\Big(\frac{(t+1)\bar{X}_{t+1}-X_{t+1}}{t}=\bar{X}_t\Big)$$
$$= \frac{1}{Z}\Big[(\kappa_t+1)\kappa_t X_{t+1}X_{t+1}^T-(t+1)^2\kappa_t\bar{X}_{t+1}\bar{X}_{t+1}^T$$
$$+t^2(\kappa_t+1)\Big(\frac{(t+1)\bar{X}_{t+1}-X_{t+1}}{t}\Big)\Big(\frac{(t+1)\bar{X}_{t+1}-X_{t+1}}{t}\Big)^T$$
$$-2\kappa(t+1)\kappa_t\mu_0\bar{X}_{t+1}^T+2\kappa t(\kappa_t+1)\Big(\frac{(t+1)\bar{X}_{t+1}-X_{t+1}}{t}\Big)$$
$$+\kappa^2\mu_0\mu_0^T\Big] \tag{18}$$

where,

$$Z = (\kappa+t+1)(\kappa+t)$$
$$= \frac{1}{Z}\Big[(\kappa_t+1)^2X_{t+1}X_{t+1}^T+(t+1)^2\bar{X}_{t+1}\bar{X}_{t+1}^T+\kappa^2\mu_0\mu_0^T$$
$$-2(\kappa_t+1)(t+1)\bar{X}_{t+1}X_{t+1}^T$$
$$+2\kappa(t+1)\mu_0\bar{X}_{t+1}-2\kappa(\kappa_t+1)\mu_0X_{t+1}^T\Big]$$
$$= \frac{1}{Z}\Big[(\kappa_t+1)X_{t+1}-(t+1)\bar{X}_{t+1}-\kappa\mu_0\Big]$$
$$\times\Big[(\kappa_t+1)X_{t+1}-(t+1)\bar{X}_{t+1}-\kappa\mu_0\Big]^T$$
$$= \frac{\kappa_t(X_{t+1}-\mu_t)(X_{t+1}-\mu_t)^T}{\kappa_t+1} \tag{19}$$

REFERENCES

[1] "Ericsson mobility report: 70 percent of world's population using smartphones by 2020," https://www.ericsson.com/news/1925907, June 2015.

[2] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[3] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*. Springer, 2004, pp. 1–17.

[4] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," in *International Conference on Pervasive Computing*. Springer, 2010, pp. 319–336.

[5] M. Ermes, J. Parkka, and L. Cluitmans, "Advancing from offline to online activity recognition with wearable sensors," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2008, pp. 4451–4454.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *J. UCS*, vol. 19, no. 9, pp. 1295–1314, 2013.

[7] ——, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International Workshop on Ambient Assisted Living*. Springer, 2012, pp. 216–223.

[8] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2011, pp. 289–296.

[9] D. Vishwakarma, P. Rawat, and R. Kapoor, "Human activity recognition using gabor wavelet transform and ridgelet transform," *Procedia Computer Science*, vol. 57, pp. 630–636, 2015.

[10] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[11] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimedia Tools and Applications*, pp. 1–14, 2015.

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[13] J.-L. Reyes-Ortiz, L. Oneto, A. Ghio, A. Sama, D. Anguita, and X. Parra, "Human activity recognition on smartphones with awareness of basic activities and postural transitions," in *International Conference on Artificial Neural Networks*. Springer, 2014, pp. 177–184.

[14] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[15] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An introduction to statistical learning," vol. 6, 2013.

[16] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[17] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *Architecture of computing systems (ARCS), 2010 23rd international conference on*. VDE, 2010, pp. 1–10.

[18] A. Mannini and A. M. Sabatini, “Machine learning methods for classifying human physical activity from on-body accelerometers,” *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[19] R. P. Adams and D. J. MacKay, “Bayesian online changepoint detection,” *arXiv preprint arXiv:0710.3742*, 2007.

[20] P. A. Tobias and D. Trindade, *Applied reliability*. CRC Press, 2011.

[21] J.-L. Reyes-Ortiz, L. Oneto, S. Albert, X. Parra, and D. Anguita, “Transition-aware human activity recognition using smartphones,” *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[22] E. Keogh, S. Chu, D. Hart, and M. Pazzani, “An online algorithm for segmenting time series,” in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 289–296.