An Adaptive Approach for Online Segmentation of Multi-Dimensional Mobile Data

Tian Guo
EPFL, Switzerland
tian.guo@epfl.ch

Zhixian Yan
EPFL, Switzerland
zhixian.yan@epfl.ch

Karl Aberer
EPFL, Switzerland
karl.aberer@epfl.ch

ABSTRACT

With increasing availability of mobile sensing devices including smartphones, online mobile data segmentation becomes an important topic in reconstructing and understanding mobile data. Traditional approaches like online time series segmentation either use a fixed model or only apply an adaptive model on one dimensional data; it turns out that such methods are not very applicable to build online segmentation for multiple dimensional mobile sensor data (e.g., 3D accelerometer or 11 dimension features like 'mean', 'variance', 'covariance', 'magnitude', etc).

In this paper, we design an adaptive model for segmenting real-time accelerometer data from smartphones, which is able to (a) dynamically select suitable dimensions to build a model, and (b) adaptively pick up a proper model. In addition to using the traditional residual-style regression errors to evaluate time series segmentation, we design a rich metric to evaluate mobile data segmentation results, including (1) traditional regression error, (2) information retrieval style measurements (i.e., precision, recall, F-measure), and (3) segmentation time delay.

Categories and Subject Descriptors

H.2 [**Database Management**]: Database Applications—*Spatial databases and GIS, Data mining*; H.3 [**Information Systems**]: Information Storage and Retrieval—*On-line Information Services*

General Terms

Algorithms, Performance

Keywords

online segmentation, mobile data mining, feature selection, adaptive model creation

1. INTRODUCTION

Recently, smartphones have became ubiquitous mobile sensing devices that have increasing abilities in computation (more processing powers, extra memory & storage, efficient networking) and sensing (with new embedded rich sensors like GPS, accelerometer, gyrospace). Such smartphones based sensing starts to establish a new paradigm of people-centric mobile sensing [4]. We are now facing research challenges in dealing with such large scale real-life mobile data from smartphones.

To gain a meaningful data understanding from such mobile sensing data, one of the major tasks in this area is to divide the long sequence of mobile sensing records (e.g., GPS, accelerometer, WiFi, etc.) into a set of individual segments. Each segment is corresponding to a "specific" concept or activity (e.g., one segment is at home and the next one is in outside; one segment is on "walking", while the subsequence one is on "cycling"). Because of such meaningful understanding, the mobile data segmentation has received a lot of attention recently in various mobile sensors, e.g., GPS-based trajectory segmentation [1, 3, 18, 15], accelerometer-based motion segmentation [7, 6, 14].

In this paper, we focus on designing efficient online segmentation method for mobile sensor streams. We identify following shortcomings for existing online segmentation methods: (1) segmentation is typically based on a uniform model, e.g., a linear regression model [8], a piecewise model [10], a high-order polynomial model [5] ; (2)as uniform models typically cannot have always good performance, there are some adaptive models (e.g., dynamic Auto-regression [2]), but all of these are for 1-D time series data; (3)segmentation only focuses on minimizing the errors of regression models (e.g., sum of residuals); however, different from segmenting traditional time series (signal data), minimizing errors is not always the main objective, but dividing the long sequence of mobility data into relatively independent segments, where each segment has a separate semantic meanings (e.g., home vs. office, sitting vs. standing). The detailed related work will be provided in Section 2.

To overcome these disadvantages in current literature on online segmentation of mobility data, this paper designs a novel adaptive approach for online segmentation of multi-dimensional mobile data. The detailed contributions of this paper are as follows:

a) *Adaptive Dimension Selection*: After generating a set of statistical features from the raw accelerometer sensor data [16], this adaptive model is able to automatically select suitable features to create a segmentation model. We design PCA (Principal Component Analysis) and PCA-enhanced feature selection strategies.
b) *Adaptive Model Selection*: During the procedure of on-

MobiDE '12, May 20, 2012 Scottsdale, Arizona, USA

line segmentation, our adaptive approach is able to select a suitable model from multiple model candidates. In this paper, we focus on applying multiple polynomial models with different orders, e.g., linear, quadratic, or cubic regression.

c) *Rich Evaluation Metrics*: Traditional metrics for evaluating time series segmentation are purely based on regression errors. In this paper, we propose rich evaluation metrics, considering regression errors, IR-based metrics (precision, recall, and F-measure), and segmentation boundary (delay).

d) *Real-Life Mobile Data Validation*: Finally, we collect real-life mobile data using the accelerometer data from smartphones, provide ground-truth tags and semantically evaluate our adaptive approach with rich metrics.

The detailed structure of this paper is organized as follows: after the introduction, Section 2 summarizes existing related works; Section 3 describes the preliminaries for the problem of online segmentation of mobile data; Section 4 studies different strategies of feature/dimension selection; whereas Section 5 focuses on adaptive model selection; in Section 6, we experimentally evaluate our approach. Finally, Section 7 includes concluding remarks and points to future works.

2. RELATED WORK

In this section, we overview the three main topics related to this paper: (1) trajectory segmentation, (2) time series segmentation, and (3) online adaptive model creation.

Trajectory Segmentation: Trajectory segmentation is originally from processing video-based motion streams (e.g., [13]), where the input data is the visual tracking (images of tracking sequence of object movement). For mobile data (participially GPS) based trajectory segmentation, the objective is to divide a long GPS trace into several sub-parts (sometimes called "trajectory episodes") and enable to further assign semantic annotations for each episode, e.g., transportation modes for the move episodes and activities for the stop episodes [17]. Recently, a set of trajectory segmentation methods are proposed, such as using velocity [19], change point detection [20], more generic spatio-temporal criteria like location, heading, speed [3]. As GPS is only one type of mobile sensors that are quite energy-hungry in smartphones. In this paper, we more focus on analyzing accelerometer sensor based mobile data segmentation, which is more applicable to real-time applications as it is the most commonly-used and low-energy smartphone sensor.

Time Series Segmentation: Time series segmentation is a hot topic in many areas such as signal processing, data mining, and applications (e.g., data modeling and forecasting) of financial & environmental data. According to well-known time series segmentation studies like [8, 9, 7], the segmentation methods are divided into three categories, i.e., *top-down*, *bottom-up*, and *sliding window*. The *top-down* and *bottom-up* methods typically have sound segmentation performance but are offline & inefficient for processing mobile streams; whilst *sliding window* can deal with real-time sequential data, but the performance is poor. To achieve a good tradeoff between offline high-accuracy and online efficient-computation, a couple of "hybrid" methods are proposed, such as SWAB (*Sliding Window And Bottom-up*) in [9], FSW (*Feasible Space Window*) & SFSW (*Stepwise FSW*) [12], SwiftSeg (a polynomial approximation of a time series in either sliding or growing windows) [5]. However, these online segmentation methods apply a model, which can generally work well for traditional time series data (e.g., stocks in finance and temperature in environment), but not for mobile data which typically is more non-stationary as generated in people's daily life. In this paper, we design an adaptive model in order to gain better segmentation performance for mobile data.

Adaptive Modeling for Segmentation: Recently, researchers start to build adaptive models for time series predication and segmentation. In [2], different degrees of AR (Autoregression) models are adaptively selected for online time series prediction, which aim at reducing the communication cost in wireless sensor network. In [11], ARMA (Autoregression Moving Average) and polynomial models are combined together for real-time time series. In [10], an adaptive model piecewise linear segmentation (mixing both constant and linear function) is designed for time series segmentation. Nevertheless, all of these adaptive models are only designed for one-dimensional time series data; in our paper, we will design an adaptive model for multiple dimensional mobile data.

3. PRELIMINARIES

In this section, we provide the problem statement of mobile data segmentation using the accelerometer sensor embedded in smartphones, and offer a rich metric for evaluating segmentation performance.

3.1 Problem Statement

The problem of "mobile data segmentation" in this paper is to automatically divide real-life collected accelerometer data (e.g., collected by the embedded sensor in smartphones) into several segments, where the data in each segment is in some sense homogeneous (e.g., doing a single activity like "siting", "standing", "running") and the data between two neighboring segments are different (i.e., belong to different activities). At the top of Fig. 1, we observe the three-dimensional accelerometer data, which is called "Raw Accelerometer Stream".

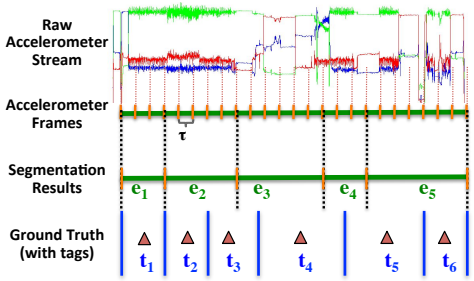

Figure 1: Segmentation of Accelerometer Stream

Definition 1 (Raw Accelerometer Stream - $\mathcal{A}$). *A sequence of data points recording acceleration in 3 dimensions, i.e.* $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, *where* $a_i = (x, y, z, t)$ *is a tuple with accelerations* (x, y, z) *and timestamp* (t).

We do not segment the raw accelerometer stream directly based on each record a of the raw accelerometer sampling stream $\mathcal{A}$, but based on each group of accelerometer records in a time frame τ (see Fig. 1), where τ is typically small (e.g., 5 secs or 10 secs). There are two advantages of segmenting on the frames τ rather than the raw accelerometer stream directly: (1) *Robust to outliers* – As our data collection is in

the naturalized setting where people uses the mobile phones without any restrictions in their daily life. The accelerometer data is much sensitive to user motions and outliers, e.g., changing phone positions, talking with phone when walking; therefore, the neighboring record a can vary a lot while the user is continuing with the same activity. The group values of a can reduce such sensitivity, which in turn to achieve a robust segmentation algorithm using accelerometer frames. (2) *Efficient segmentation* – The raw data stream typically is very huge, as accelerometer data sampling usually has a very high frequency (e.g., 30 records per second). With accelerometer frame, we can obtain efficient segmentation as the size of the accelerometer frames is significantly less than the raw accelerometer. For example, in our experiment, accelerometer is sampling at 30 Hz, by using frame size of 5 secs ($\tau = 5$), the ratio between accelerometer frames and accelerometer records is $\frac{1}{5*30}$, a remarkable compression before data segmentation.

DEFINITION 2 (ACCELEROMETER FRAME - $A^{(\tau)}$). *A set of continuous accelerometer records, i.e.* $A^{(\tau)} = \{a_{i+1}, \ldots, a_{i+k}\}$ *in time duration* τ. *Basic features are calculated to describe the frame characteristics* $\langle f_1, \cdots, f_l \rangle$, *where* f_j *can be time domain features like mean, variance of the three axis* (x, y, z) *and frequency domain features like energy, entropy, and magnitude etc.*

In [16], over 70 features are calculated for each accelerometer frame that is used for offline activity mining. In this paper, we apply a small set of features for real-time segmentation – 22 features (*mean_x*, *mean_y*, *mean_z*, *mean_mag*, *mag_mean*, *var_x*, *var_y*, *var_z*, *cov_xy*, *cov_xz*, *cov_yz*, *corr_xy*, *corr_xz*, *corr_yz*, *energy_x*, *energy_y*, *energy_z*, *energy_mag*, *entropy_x*, *entropy_y*, *entropy_z*, *entropy_mag*). Therefore, raw mobile data of accelerometer becomes 22 dimensional mobile streams [1]. The problem of mobile data segmentation is: *Given incoming accelerometer frames with 22 dimensional features, the segmentation algorithm is to identify meaningful episodes on the fly.* As shown in Fig. 1, the segmentation results from the accelerometer frames are five episodes (e_1 to e_5).

3.2 Segmentation Metrics

In traditional time series segmentation, the evaluation metric is largely based on the modeling/regression errors (e.g., *RSS - Residual Sum of Squares*). Therefore, the objective of segmentation algorithms is to minimize the total RSS for all of individual segments. This is good for evaluating the signal style time series (e.g., financial data, environmental measurements), but not always suitable for evaluating mobile data stream, which can have the ground-truth segments that can be provided by users.

In our Nokia phone platform, user can provide activity tag in their daily life in real time via the phone. The micro activity tag can be sitting, standing, jogging, walking, climbing etc. As shown in Fig. 1, there are six tags. Such ground truth tags can validate the segmentation results. We design the measurement based on the widely used metrics in information retrieval (i.e., *precision* and *recall*) to measure the segmentation accuracy, as shown in Formula 1. Based on this definition, the optimal segmentation is in the final result where each segment has one and only one tag inside, i.e., both precision and recall are 100%.

[1] Hereby, 'feature' and 'dimension' share the same meanings, and we do not distinguish them in this paper.

$$\begin{aligned} precision &= \frac{\#segments\ with\ tag}{\#\mathcal{E}_A} \\ recall &= \frac{\#segments\ with\ tag}{\#total\ tags} \\ f{-}measure &= \frac{2 * precision * recall}{precision + recall} \end{aligned} \tag{1}$$

The accelerometer segmentation example in Fig. 1 has five segmented episodes, where four episodes have user tags inside (e_1, e_2, e_3, e_5) and two episodes have more than one tags. According to Formula 1, the precision is $\frac{4}{5} = 80\%$, recall is $\frac{4}{6} = 66.7\%$, and $f{-}measure$ is $\frac{2*4/6*4/5}{4/6+4/5} = 72.7\%$.

In the measurement of Formula 1, all the segments are considered equally, similar as all the tags. We can further define the weighted (w) precision and recall (see Formula 2), where the length of segment and the tag interval distance are also measured to validate the segmentation result.

$$\begin{aligned} precision^{(w)} &= \frac{\Sigma\{length\ of\ segment\ with\ tag\}}{\Sigma\{length\ of\ e_i\}} \\ recall^{(w)} &= \frac{\Sigma\{tag\ interval\ with\ division\ inside\}}{\Sigma all\ tag\ intervals} \\ f{-}measure^{(w)} &= \frac{2 * precision^{(w)} * recall^{(w)}}{precision^{(w)} + recall^{(w)}} \end{aligned} \tag{2}$$

In such case, the weight measurement of the segmentation accuracy in Fig. 1 is: $precision^{(w)} = \frac{|e_1|+|e_2|+|e_4|+|e_5|}{\sum_{i=1}^{t} |e_i|}$, and $recall^{(w)} = \frac{|t_1t_2|+|t_3t_4|+|t_4t_5|}{\sum_{i=1}^{5} |t_it_{i+1}|}$, where $|t_it_{i+1}|$ is the distance interval between the two neighboring tags.

In addition to *precision* and *recall*, we also evaluate segmentation performance in terms of *detection delay*, i.e., the "time gap" between the segmented division point (i.e., the dotted division line in Fig. 1) and the nearly ground-truth division (i.e., the blue division line in Fig. 1). This metric can evaluate how the algorithm can detect the segments promptly (neither too early nor too late).

4. ADAPTIVE FEATURE SELECTION FOR SEGMENTATION

In multi-dimensional (i.e., features) time series, selecting a suitable set of features for building segmentation is a critical step that could affect the segmentation results significantly. In this section, we study different feature selection strategies when building online mobile data segmentation.

4.1 Segmentation with Fixed Features

Now, we discuss how to design an online segmentation algorithm with manually fixed features. We extend the SWAB (Sliding window and bottom up) online segmentation algorithm [8][9], and build a multi-dimensional time series segmentation of accelerometer units. The reasons of extending SWAB includes: (1) online segmentation which can be applied in real-time systems; (2) efficient segmentation which can be easily transplanted in the smartphone platform, as the time complexity is linear and much faster compared to any polynomial or dynamic programming based sophisticated methods.

Alg. 1 briefly shows the procedure of such real-time accelerometer data segmentation: (1) divide the data stream

into frames and calculate the features of each frame; (2) manually choose a feature set (F_s) from complete feature set (F) for building online regression in the window; if the regression error is less than a given threshold (ϵ), then the window continuously grows, otherwise a new segment appears.

Algorithm 1: Segmentation of accelerometer stream by a manually selected feature set

Input: the raw accelerometer stream $\mathcal{A}$, a regression error bound (ϵ), manual feature index (idx)
Output: a sequence of accelerometer episode $\mathcal{E}_A$

```
1  begin
2      /* step 1: divde stream into frame */
3      A → {a_1^(τ), ··· , a_m^(τ)}, where τ is the time unit.
4      /* step 2: fix feature set */
5      F_s = selectFeatures (F, idx);
6      /* step 3: online segmentation with incoming frame */
7      E_A ← ∅; W ← ∅;
8      while hasNextFrame(A) do
9          frame = nextFrame(A);
10         T = featureExtract(frame, F_s);
11         err = linearRegressionModel(W ∪ T);
12         if err < ε then
13             //grow window with next frame;
14             W ← W ∪ T; continue;
15         else
16             // a new segment appears
17             E_A.addOutSegment(W);
18             W ← T;
19     calculate metrics (f-measure) on real tags (see Formula 1);
20     return E_A (with f-measure);
```

Typically the feature dimension is large (e.g., our accelerometer data has 22 features), so manual feature selection is very bias and burdensome to evaluate. Therefore, in the coming section, we discussion better feature selection strategy.

4.2 Segmentation with PCA Feature Selection

The PCA (Principle Component Analysis) is a classical feature selection and dimension reduction method in dealing with multivariate time series data. PCA tries to find a linear transformation matrix U to project original data to lower dimensional space, where the correlation between new dimensions are minimized. Through PCA, we can get an eigenvalue vector $V = \{v_1, \cdots, v_n\}$, where v_i corresponds to the i^{th} column of transformation matrix U. The rational of PCA based feature selection is to select k orthogonal bases with the biggest k eigenvalues in V.

Different from the manual feature selection in Alg. 1, PCA-based feature selection for online segmentation can automatically identify the feature index idx for building the regression model. Alg. 2 briefly shows the new function of $selectFeaturesPCA$, which replaces the manual function $selectFeatures$ with given index (idx) in the online segmentation algorithm.

Segmentation with PCA based feature selection can gain better performance in segmentation. However, there are also some drawbacks for directly using PCA, e.g., : (a) For every new incoming data, it should take linear transformation. This operation is still heavy for online processing of time series with many features. (b) Since the model is based on the projected space, if we want to retrieve historical data through the model, the reversed transformation is needed to get the real data in original space. This process may bring computation error and is also time-consuming. (c) The principle of PCA feature selection is to select a set of orthogonal base to transform data to new projected space. This

Algorithm 2: selectFeaturesPCA

Input: W // current window including accelerometer frames, a PCA component threshold(ϵ_{pca})
Output: a new W' with selected features

```
1  begin
2      {V, U} = PCA(W)
3      /*eigenvalue vector V = {v_1, ··· , v_n} in decreasing order
       and transformation matrix U */
4      select n_s that satisfies Σ_{i=1}^{n_s} v_i / Σ_{i=1}^{n} v_i ≥ ε_pca.
5      idx = {u_1, ··· , u_{n_s}}; // construct feature selection index
6      W' = W × idx; // a new matrix with selected features
7      return W'
```

transformation maximizes the variance of every dimension in projected space. The bigger the variance is, the larger the fluctuation of data on this dimension is. As a result, it is hard to distinguish whether the data changes are caused by the beginning of a new segment or internal impulse within current segment.

4.3 Segmentation with Enhanced Feature Selection (PCA$^+$)

Based on above analysis, an ideal online feature selection algorithm should be able to (1) perceive the hidden structure of them and eliminate redundant features to decrease computation complexity; (2) for each segment, the feature selection is capable of automatically selecting the most representative features used for segment modeling; (3) Within the same segment, data variation on these chosen features should be stable to avoid mis-segmentation. Between different segments, the changes ought to be salient enough so that model based segmentation can detect it. We define these features as *Stable and Salient Feature* (SSF). Therefore, the objective of online feature selection is to adaptively adjust SSF features when doing online segmentation.

The PCA still makes sense for discovering underlying structure and variation pattern of data. We will explore some properties of PCA and hierarchical clustering.

The PCA is a significant tool to find SSF and hidden structure of features. After making PCA on data of the current interval, we can get the eigenvalue vector $V = \{v_1, \cdots, v_n\}$ and projection matrix $U = \{u_1, \cdots, u_n\} = \{r_1^T, \cdots, r_n^T\}$. The row vector r_i of U has two hidden meanings.

(1)The $r_{i,j}$ reflects *i-th* original dimension's contribution to the *j-th* one of new projected space. For example, as the elements in V are in descending order, the biggest one is the first PC(principle component). If $r_{k,1}$ is the largest one of row vector r_k, the *k-th* dimension of original data contributes most to first PC and the data on the *k-th* dimension of orginal space also has biggest variance. This will help to find the proper features for segmentation.

(2)The correlated dimensions would have similar r_i. For example, for two linear dependent features f_a and f_b, $\|r_a - r_b\| \approx 0$. This property is beneficial for eliminating the redundant dimensions, selecting the most representative dimensions and simplifying the complexity of modeling.

Above analysis indicates that we can exact enough information from r_i upon variance of each feature and correlation between features. In order to automatically extract SSF and other representative features, hierarchical clustering is introduced here.

Once we get row vector $\{r_1^T, \cdots, r_n^T\}$, hierarhcial clustering algorithm is able to divide them into different groups

$\{g_1, \cdots, g_l\}$ each of which consists of correlated features. In view of above analysis on conventional PCA feature selection, the features that contribute most to the first PC is not suitable for time series segmentation, thus the cluster to which the $u_{1,i}$ with the maximum value belongs is eliminated. Based on the experimental observation, the feature corresponding to v_{i2} with biggest value among second PC u_2 possess tolerable variance within segment and salient enough change on the boundary of different segments, so it accords with SSF's definition. We extract the feature s with biggest value $v_{s,2}$ in u_2 as SSF. For other clusters, we just simply select the features corresponding to the center element of every cluster. The model used for segmentation is usually the explicit mulitvariate model. The defintion of SSF inspires us to make it as the dependent variable of model, because the data variation on SSF is easy and robust for precise segmentation. In this way, we have constructed a enhanced feature subset for segmentation. See the pseudocode of enhanced feature selection in Alg. 3

Algorithm 3: EnhanFeaSel($\{t_i, \cdots, t_j\}$)

Input: the raw accelerometer stream $\mathcal{A}$
Output: selected feature set F_s

```
1  begin
2      /* step 1: divide stream into units*/
3      A → {a_1^(τ), ··· , a_m^(τ)}, where τ is the time unit.
4      for ∀a_i ∈ A, calculate the values of every feature in F to get
       X = {x_1, ··· , x_n}
5      /* step 2: apply PCA */
6      {V, U} = PCA(X)
7      /*eigenvalue vector V = {v_1, ··· , v_n} in decreasing order
       and transformation matrix U */
8      /* step 3: apply hierarchical clustering */
9      G = {g_1, ··· , g_k} = hierClu (r_1, ··· , r_n)
10     /*hieClu() is the hierarchical clustering function. */
11     g_e= cluMap (max(u_1))
12     /*cluMap() is group mapping function and max() returns
       the row number of maximum element in vector u_1 */
13     G' = G − g_e
14     for ∀g_i ∈ G', draw f'_j = center(g_i)
15     /* function center() return the center element in cluter */
```

5. ADAPTIVE MODEL SELECTION FOR ONLINE SEGMENTATION

Traditional online segmentation methods typically apply uniform model for determining segments. This is bias to the model and overlooks the variation of data's hidden structure. In this section, we design an adaptive method that is not only able to adaptive select features but also can pick a suitable model during the procedure of online segmentation.

5.1 Model Selection Criteria

Model selection is able to choose a suitable model for each segment. Given a set of model candidates $\mathcal{M} = \{M_1, \cdots, M_m\}$, the online segmentation algorithm is able to adaptively select the most suitable model for the current segmenting window. In this paper, we consider three polynomial models (e.g., linear, quadratic, and cubic). To evaluate the model, we define n_o as the model order (e.g., 1 for linear, 2 for quadratic, and 3 for cubic), and n_v as the number of variables in model.

For modeling current segment, we need to design a criteria to select proper model. We need to concern two aspects in selecting the model, i.e., *model precision* and *model cost*.

- *Model Precision*: Only when the modeling error is small enough, the change incurred by new segment's data can be detected in real time. In addition, the reconstructed data from this model may address the accuracy requirement of information retrieval.
- *Model Cost*: This is the computation complexity for modeling. As segmentation algorithm need will run in online manner, model complexity affects the performance of realtime processing.

In general, the more accurate the model is, the larger computation cost is required. We note $\mathcal{MP}$ as model precision and $\mathcal{MC}$ as model cost. The model selection becomes a multi-objective optimization problem, which can be simplified as a linear combination as follows:

$$\underset{M_i \in \mathcal{M}}{\operatorname{argmin}} \quad \mathcal{MR}(M_i) + \delta \times \mathcal{MC}(M_i) \tag{3}$$

Model precision (MP) can be approximated as the regression errors (e.g., RSS). In segmentation, RSS is monotonous increasing with new incoming data point. We apply the average RSS for each segment. Suppose R is segment of modeling. t_i is the data point in R and n_r is the number of points in R. y_i is the dependent-variable of t_i and x_i is the independent-variables of t_i. Then the MP of M_i can be defined as:

$$\mathcal{MP}(M_i) = \{\sum_{t_i \in R} (y_i - M_i(x_i))^2\}/n_r \tag{4}$$

Regarding model cost (MC), we apply the time complexity analysis. For any regression model with n_o order and n_v variables, the time complexity for modeling is $O(n_o * n_v)$. In addition, we analyze the coefficients of the computed model, i.e., $C = \{c_1, \cdots, c_t\}$. Based on the observation of model fitting experiment, there are some much smaller coefficients in C, which means that the associated items contribute very little to the model. In this sense, we define subset

$$C' = \{c_i \mid c_i \in C, c_i \leq \epsilon\}, 0 \leq \epsilon \leq 1. \tag{5}$$

where only the elements in $C - C'$ make sense for the dynamic computation cost.

In sum, the overall computation cost of model consists of *constant* and *dynamic* cost, as follows:

$$\mathcal{MC}(M_i) = n_o \times n_v + \sum_{c_i \in C - C'} n_i^t \tag{6}$$

where n_i^t is the number of variables in the i^{th} item in the polynomial model.

5.2 Adaptive Feature and Model Selection for Segmentation

This section provides our adaptive method that is able to adaptively select both features and model for online segmentation. The online segmentation decision is to determine whether the new coming data point belongs to current or a new segment. This is an optimization problem. The optimal segmentation on time series is to find the points $(i.e, p_1, p_2, \cdots, p_\infty)$ to divide time series into different segments $(i.e, R_1, R_2, \cdots, R_\infty)$ such that the sum of individual

segment's RSS is minimized as follows (see Formula 7).

$$\underset{p_0,\cdots,p_q}{\operatorname{argmin}} \quad RSS(p_0,\cdots,p_q) = \sum_{i=1}^{q}\sum_{j=p_i}^{p_{i+1}-1}(M_i(x_j)-y_j)^2 \quad (7)$$

In theory, this problem can be solved by dynamic programming (DP) and achieve an optimal solution for offline segmentation. For online segmentation, Alg. 1 has designed a heuristic and near optimal solution. Furthermore, Section 4 provided three feature selection strategies, i.e., manual, PCA and PCA$^+$ methods. In this section, we combine feature selection together model selection.

Assume the chosen feature subset and model of current segment R_c is $\widehat{F} = \{f_1, \cdots, f_s\}$ and M_c. t_i is the new coming data point. x_i and y_i are separately the independent and dependent variable parts of t_i. The distribution of data points in space constructed by approximate optimal feature set has the following attribute that for data point $t_j \in R_i$, $(M_c(x_j) - y_j)^2 < \delta$ while for the data t_e from new segment R_{c+1}, $(M_c(x_e) - y_e)^2 \gg \delta$. Fortunately, adaptive model creation offers heuristic information about δ. The precision of model reflects the even deviation of data points from fitting model. If a data point's residual error is much bigger than the precision of model, it is reasonable to consider it as a segmentation point. After adaptive model creation for a new segment, set the precision of the chosen model as ξ. Then we set δ as $\beta \times \xi$, where parameter $\beta \geq 1$ is tolerance factor for noisy data within the segment.

We will further analyze the performance of segmentation. The segmentation method will get a sequence of division points $(p_0, p_1, \cdots, p_q)$ and corresponding segments $(R_0, R_1, \cdots, R_q)$ where p_i is the staring point of segment R_i. Assume that the number of points in each segment is $(n_0^{'}, \cdots, n_q^{'})$. Since division point p_{q+1} is not detected, the number of points in segment R_q is still increasing.

$$\begin{aligned}\underset{p_0,\cdots,p_q}{\operatorname{argmin}} RSS(p_0,\cdots,p_q) &= \sum_{i=0}^{q}\sum_{j=p_i}^{p_{i+1}-1}(M_i(x_j)-y_j)^2 \\ &\leq \sum_{i=1}^{q-1}(\delta_i \star n_i^{'}) + \delta_q \star n_q^{'} \\ &\leq (\max_{i\in(0,q-1)}\delta_i) \star \sum_{i=0}^{q-1} n_i^{'} + \delta_q \star n_q^{'}\end{aligned} \quad (8)$$

From above induction, the expression $(\max_{i\in(0,p-1)}\delta_i) \star \sum_{i=0}^{p-1} n_i^{'}$ is the RSS's upper bound of all the fixed segments before R_q. The expression $\delta_q \star n_q^{'}$ is the RSS of current segment. In segment R_i, in the $j-th$ step of new coming data processing, renewed δ_i is denoted as δ_i^j. Through mathmatical induction, we can prove that all δ_i^j is bound by $\beta_i * \xi_i$, so that

$$\underset{p_0,\cdots,p_q}{\operatorname{argmin}} RSS(p_0,\cdots,p_q) \leq (\max_{i\in(0,q)}\beta_i \star \xi_i) \star (\sum_{i=0}^{q-1} n_i^{'} + n_q^{'}) \quad (9)$$

Above formula indicates that RSS of all segments presents linear increasing relation with number of data points and the average RSS of every segment is bounded by constant $\beta_i \times \xi_i$.

Now we have designed every component of online adaptive segmentation for time series. See Alg.4 for the complete algorithm description.

Algorithm 4: onlineAdaFeaModSeg(t^i)

```
Input: t_i
Output: whether t_i is segmentation point or not
1  begin
2      if IsNewSeg(t^i, δ_c, n_c) = True then
3          n_c = n_c + 1. return False
4      else
5          n_c = 1 continue reading subsequential data {t_i, ··· , t_j}
           as the data set for model initialization.
6          F_s = ApproxiOptFeaSel ( {t_i, ··· , t_j})
7          forall the M_i in M = {M_1, M_2, M_3} do
8              if MP(M_i) + MC(M_i) > optiV then
9                  optiV=MP(M_i) + MC(M_i) k_opt=i;
10         Use M_{k_opt} to model data of {t_i, ··· , t_j} on F_s.
11         ξ= MP(M_i)
12         return True /* Generate a new segment */
```

For every new data point, the segmentation method judges whether it is a division point. If yes, continue reading new data point. If no, make feature selection and model creation for the new segment.

6. EXPERIMENT

This section presents our experimental results of adaptive feature selection and model creation algorithm. Based on the ground-truth data, we apply rich metrics (i.e., RSS, f-measure, segmentation delay) to evaluate different segmentation methods. RSS is widely used in traditional time series segmentation from the viewpoint of data fitting; whilst f-measure and segmentation delay is more meaningful to evaluate the performance of segmenting the mobile data that is generated during people's different activities (e.g., sitting, standing, walking).

6.1 Experimental Setup

In our experiment, we developed Python scripts in the Nokia N95 phones carried by our subjects as their regular phones. There was no restriction on (1) where the phone should be kept in the body (2) how the phone should be used. The subjects were further requested to use the phone in exactly the way they would use a normal daily phone. Therefore, such data collection is purely in a naturalized setting which can cause more noisy data and validate the robustness of our adaptive segmentation algorithm. The python script samples the accelerometer at 30Hz (i.e., 30 samples/second).

6.2 Results of Adaptive Feature Selection

We firstly analyze online segmentation performance of using different feature selection strategies (i.e., *manual*, PCA, PCA^+) on a fixed model. For the *manual* feature selection, we exhaustively select three features, and choose the best result as the performance of *manual* based feature selection. For the fixed model, we tested three models, i.e., linear (*order*-1), quadratic (*order*-2), cubic (*order*-3).

Fig. 2 shows the RSS modeling errors for three feature strategies (i.e., *manual*, PCA, PCA^+) combined with three fixed models (i.e., *order*-1, *order*-2, *order*-3). With the order growing, both *manual* and PCA feature selection can achieve a decreasing RSS trend. However this doesn't work for *order*-3 with PCA^+. Similarly, PCA^+ achieves better performance compared with *manual* and PCA at low orders (*order*-1 & *order*-2) but not for *order*-3. The features chosen by PCA^+ has relatively stable data distribution within the

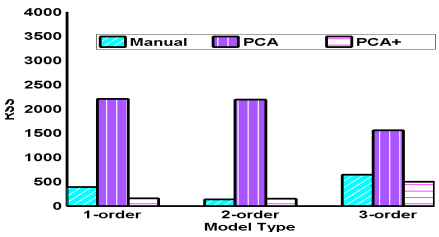

Figure 2: Results comparison (RSS)

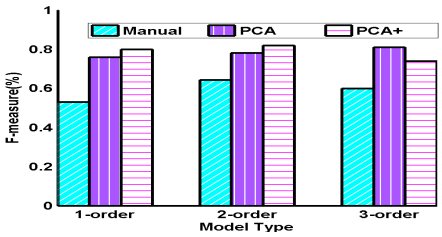

Figure 3: Results comparison (F-measure)

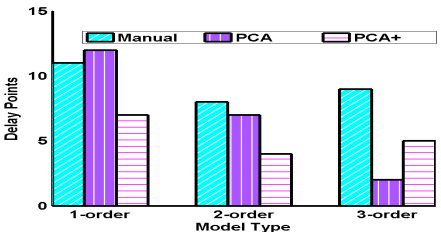

Figure 4: Results comparison (delay)

segment and data changes only between segments, therefor high order model may incur over-fitting and large RSS.

Different from the RSS errors, Fig. 3 and Fig. 4 sketch the performance from the activity segmentation perspective based on the real-life ground truth tags. We observe that PCA^+ works better at *order*-1 and *order*-2, i.e., with high *f-measure* and small segmentation *delay*. In *order*-3 based segmentation, performance deterioration occurs again for PCA^+, since the over-fitting model may lead to some mis-segmentations. Although *order*-3 model with PCA^+ feature selection are over-fitting in terms of both metrics, namely the regression metrics (i.e., *RSS*) and the IR-style metrics (i.e., *f-measure* & *delay*). The segmentation result is more robust in terms of the IR-style metrics compared with the regression metrics.

Based on the above results and analysis, complicated models (e.g., PCA^+ with *order*-3) do not always work well for all segments. This experimental evidence shows the necessary of using adaptive modeling for online segmentation, and we will see the improvement brought by adaptive modeling in the following experiments.

6.3 Results of Adaptive Model Selection

Now, we test our fully adaptive modeling, which includes adaptive model selection when using any feature strategy.

Table 1 presents the results (all three metrics) by using the fully adaptive modeling, where models are automatically selected amongst the three candidate models (i.e., *order*-1, *order*-2, *order*-3) at each segmentation step. We compare the adaptive modeling results with the best results achieved from any fixed models. As shown in Table 1, the performance values at the left side of each element are generated by adaptive modeling, whilst the values at the right side (i.e., with star $*$) are the best values from the three fixed models. In terms of *F-measure*, we observe that adaptive modeling can gain better performance in most cases: for *Manual* and *PCA*6+, adaptive modeling is better, whilst for *PCA* the best fixed model is slightly better than adaptive modeling. Considering *RSS*, all cases adaptive modeling is better than any fixed model. With regards to the segmentation delay, adaptive modeling also overperforms all fixed models when using *Manual* and *PCA* based feature selection; but for PCA^+, adaptively modeling has larger delay at 7 compared with 4 of the best fixed model. This may be caused by the large time complexity in computing PCA^+.

Table 1: Adaptive Modeling vs. Fix models

	F-measure	Delay	RSS
Manual	0.76/0.7*	3/7*	624/712*
PCA	0.82 /0.83*	2/7*	802/1060*
PCA^+	0.9/0.83*	7/4*	78/700*

Fig. 5 explicitly shows the results of segmentation using PCA^+ based feature selection together with adaptive modeling. The three signals are the means of accelerometer data in three dimensions (i.e, $mean\_x, mean\_y, mean\_z$); the doted vertical lines indicate the segmentation results computed by the algorithm; the real vertical lines are the ground truth segmentation between two continuous activities, namely, every real segment corresponds to an activity tag act_i. From the figure, we observe that only in act_5, there is one mis-segmentation and in other segments the adaptive feature selection and modeling works well.

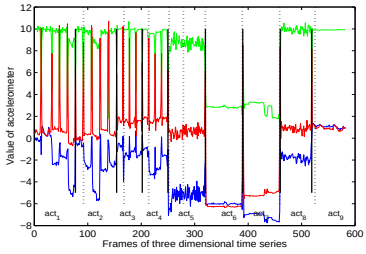

Figure 5: Segmentation results vs. ground truth tags

Additionally, Table 2 records the selected features (by PCA^+) and the selected model (by adaptive modeling) at each segment, i.e., the 10 segments corresponding to Fig. 5. We observe that at each segment, the feature selection strategy dynamically picks up suitable features sets and chooses 4-7 features from the total 22 features; in the meanwhile, the adaptive modeling automatically select the model from the two candidates (*order*-1 and *order*-2). We did not consider *order*-3, as it shows overfitting results based on our pervious experiments in Section 6.2.

Table 2: Selected Features & Model in Each Segment

Segment ID	Selected Features	Chosen Model
1	4,6,7,8,18	*order*-2
2	8,12,15,18	*order*-1
3	6,9,14,15,16,18	*order*-1
4	5,9,10,11,15,18	*order*-1
5	6,9,11,15,17,19	*order*-2
6	3,7,9,10,11,16,17	*order*-1
7	5,12,13,14,15,17,19	*order*-1
8	7,8,13,14	*order*-1
9	7,8,11,17,18,20	*order*-1
10	11,12,14,16,19,21	*order*-1

6.4 Parameter Sensitivity Analysis

Finally, we study the sensitivity of important parameters in online segmentation. This sensitivity analysis assisted us to find the best performance of each algorithm to evaluate the robustness of our adaptive feature and model selection.

The first important parameter is the total error bound (ϵ) in the segmentation algorithm. The error bound ϵ in making segmentation decision is applied in all the presented segmentation algorithms in the previous sections. We set this experiment on the simple manual feature selection and $1-order$ uniform model based segmentation to identify its influence to the final performance. Fig. 6 shows this sen-

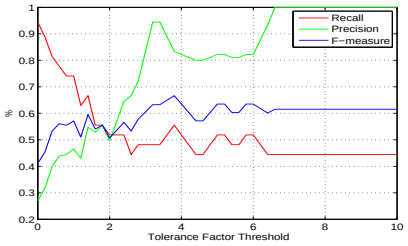

Figure 6: Total error bound (ϵ)

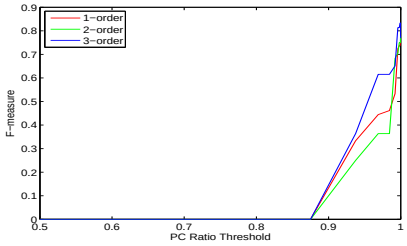

Figure 7: PC ratio (ϵ_{PCA})

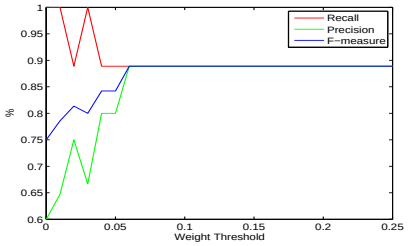

Figure 8: Model weight δ

sitivity of the results on precision, recall and f-measure. In the initial period, the low value of ϵ makes the segmentation decision sensitive to data changes, so that the recall is high while the precision is low. As the value of ϵ increases, the precision is improved and the recall becomes worse. Finally, the recall, precision and f-measure are all stable.

The second important parameter is the PC (Principal Component) ratio (ϵ_{PCA}) in applying the PCA based feature selection. Fig. 7 shows the sensitivity of ϵ_{PCA}. This experiment is the online segmentation based on PCA feature selection and $1-order$ uniform model. We only show the $F\text{-}measure$ in Fig. 7. We observed that ϵ_{PCA} only plays an effective role in a large ratio between $(0.875, 1)$. Between $(0.5, 0.875)$, none feature is chosen by the PCA based feature selection and the segmentation actually didn't work, since the first PC occupied big ratio of sum of all PCs. Then from 0.875 point, as more features are chosen, the $F\text{-}measure$ also increases. When the ratio approaches 1, the number of selected features is close to the total number of features and the variation of $F\text{-}measure$ is nearly stable.

The third parameter for sensitivity analysis is the weight parameter (δ) in model selection. In Fig. 8, from 0.065, model selection reached a balanced state and the metrics achieved optimal value. In terms of $F\text{-}measure$, even when the model weight is low, adaptive modeling still have the best results compared to any fixed models. This proves the effectiveness of our proposed algorithm.

7. CONCLUSION

In this paper, we proposed an adaptive approach on online segmentation of multi-dimensional mobile data. This approach can smartly choose features and models by approximate optimization of fitting data stream, and generate *correct* segments in a *in-situ* mode. As far as we know, this is the first work on real-time segmentation considering both adaptive models and features. We evaluated this approach using real-life datasets: continuously activity recognition by accelerometer data from mobile phones; and our experiment results demonstrated the good performance on residual error of segment modeling, segmentation precision and recall.

Our future work is to test this adaptive feature and model selection method on the smartphones. Furthermore, we will proceed pattern analysis to infer the semantic tags for each segment identified in the real time.

8. REFERENCES

[1] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. J. Keogh, and P. S. Yu. Global distance-based segmentation of trajectories. In *KDD*, pages 34–43, 2006.

[2] Y. Borgne, S. Santini, and G. Bontempi. Adaptive Model Selection for Time Series Prediction in Wireless Sensor Networks. *Signal Processing*, 87(12):3010–3020, 2007.

[3] M. Buchin, A. Driemel, M. J. van Kreveld, and V. Sacristan. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *J. Spatial Information Science*, 3(1):33–63, 2011.

[4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12:12–21, 2008.

[5] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick. Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2232–2245, 2010.

[6] E. Guenterberg, S. Ostadabbas, H. Ghasemzadeh, and R. Jafari. An automatic segmentation technique in body sensor networks based on signal energy. In *BodyNets*, pages 21:1–21:7. ICST, 2009.

[7] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. Toivonen. Time Series Segmentation for Context Recognition in Mobile Devices. In *ICDM*, pages 203–210, 2001.

[8] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An Online Algorithm for Segmenting Time Series. In *ICDM*, pages 289–296, 2001.

[9] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting Time Series: A Survey and Novel Approach. In *Data mining in Time Series Databases*, pages 1–22. Publishing Company, 2004.

[10] D. Lemire. A Better Alternative to Piecewise Linear Time Series Segmentation. In *SDM*, pages 545–550, 2007.

[11] A. Li, S. He, and Q. Zheng. Real-Time Segmenting Time Series Data. In *APWeb*, pages 178–186, 2003.

[12] X. Liu, Z. Lin, and H. Wang. Novel Online Methods for Time Series Segmentation. *IEEE TKDE*, 20(12):1616–1626, 2008.

[13] R. Mann, A. D. Jepson, and T. F. El-Maraghi. Trajectory Segmentation Using Dynamic Programming. In *ICPR (1)*, pages 331–334, 2002.

[14] L. Oudre, A. Lung-Yut-Fong, and P. Bianchi. Segmentation of Accelerometer Signals Recorded During Continuous Treadmill Walking. In *EUSIPCO*, Barcelona, Spain, 2011.

[15] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis. Segmentation and Sampling of Moving Object Trajectories based on Representativeness. *IEEE TKDE*, PP(99):1, 2011.

[16] Z. Yan, D. Chakraborty, A. Misra, H. Jeung, and K. Aberer. SAMMPLE: Detecting Semantic Indoor Activities in Practical Settings using Locomotive Signatures. In *ISWC*, 2012.

[17] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories. In *EDBT*, 2011.

[18] Z. Yan, N. Giatrakos, V. Katsikaros, N. Pelekis, and Y. Theodoridis. Setrastream: Semantic-aware trajectory construction over streaming movement data. In *SSTD*, pages 367–385, 2011.

[19] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty. A Hybrid Model and Computing Platform for Spatio-Semantic Trajectories. In *ESWC*, pages 60–75, 2010.

[20] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *WWW*, pages 791–800, 2009.