Three Dimensional Features for Human Activity Inference on Smart Phones

Chien-Hsiang Lai, Gunarto Sindoro Njoo, Wen-Chih Peng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
{ likekaito@gmail.com, gunarto.cs01g@g2.nctu.edu.tw, wcpeng@cs.nctu.edu.tw }

***Abstract*—Selecting best features for human activity inference in smart phone is a challenging task. In this paper, we tackle the problem of activity inference in smart phone by utilizing three dimensions of smart phone data: 1) Spatial-Temporal data: reflecting daily routines and semantic of locations, 2) Application: perceiving specialized apps that assist user's activities, 3) Motion: distinguishing motion-type activities from others. In addition to smart phone data, we utilize outsourced dataset to address public opinions to infer users' activities. Finally, we compare our proposed method with several common classification methods on real dataset to evaluate the effectiveness and performance of our method. Experimental results show that our approach outperformed other methods in terms of accuracy, running time, and storage efficiency.**

I. Introduction

With the technology changing every day, smart phones have been developed rapidly and become more popular in our life. According to the statistics from *we are social* [1] in 2014, world population was around 7.2 billion people, while there was around 3.6 billion smart phones in the world, i.e., one of the two people hold a smart phone. On the other hand, *Forrester research* [2] also had a marketing research showing that smart phones are integrated closely into our daily life. Moreover, nowadays, there are many applications, such as: *Google Now* [3], *Motorola Assist* [4], and *Yahoo Aviate* [5], which could infer users' activities in order to provide assistance and intelligent services. *Google Now* collects individual habits, location data, and repetitive actions to predict what user would be doing and provide more relevant and useful information. On the other hand, *Motorola Assist* can recognize some specific activities (e.g., meeting, driving, etc.) by taking advantage of several sensors, such as: accelerometer, calendar, and GPS, to prevent unexpected interrupts. For example, when driving or sleeping, *Motorola Assist* could give an assistance by filtering phone call. Differently, *Yahoo Aviate* detects users' activities and then presents relevant applications based on users' context on the mobile phone home screen.

In general, human activity inference is categorized as a classification problem in data mining field. There are some typical challenges about classification problem. First, feature selection: what features have the biggest role in the inference process. Second, feature integration: Integrating features with different characteristics. Third, performance: how to model the data to infer users' activities accurately.

Recognizing users' activities from mobile data is the main trend in the future [6], because this topic could bring many business opportunities by offering services to user, such as: customized advertisements and just-in-time information. For instance, we could promote users some coupons or discounts, which are provided by nearby store, when users are shopping or provide a real time traffic status and driving plan to users who are driving or using transportation.

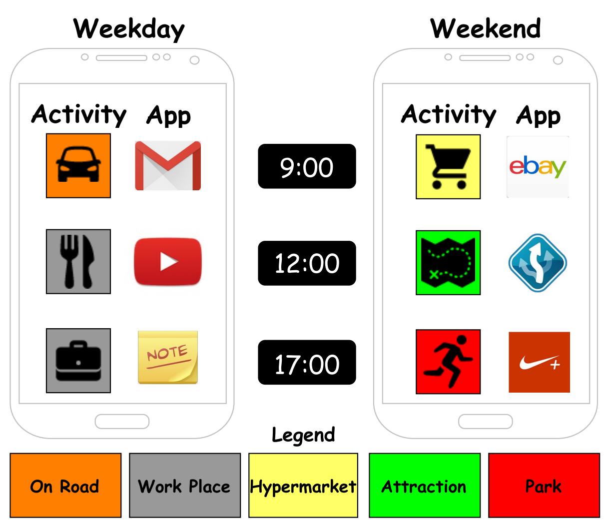


Fig. 1: An example of user daily file.

Motivating Example. Figure 1 displays an example of daily activities logs that are collected from Paul's smart phone. In the weekdays, he goes to work at 9:00 a.m. and uses Gmail to receive email notifications during the trip. At noon, he eats lunch while also listens to the music by using music player at his office. In the afternoon, he has a meeting and uses a note application to record meeting's summaries. However, there is a significant difference between his weekday's lifestyle and weekend's lifestyle. As for weekend, he goes to hypermarket for shopping in the morning. At noon, he has a small trip and uses navigation service to get the driving plan to the destination. Subsequently, the user jogs around a nearby park and uses *Nike Plus* to calculate calories and mileage. By observing Figure 1, we could discover some interesting phenomena and summarize the observations into several factors as follow.

Temporal factor. From the example above, we know that the user performs two different activities (i.e., working and dining) at the same location but each activity occurs on different time slots (i.e., lunch at 12 p.m. and work at 5 p.m.). Therefore, we argue that time is a distinctive feature for classifying user's activities.

Location factor. On the other hand, subject has completely different activities between weekday and weekend even though they are performed within the same time slot. However, we could approximately infer what the user does by observing user's location. For example, company is the place for working and hypermarket for shopping, etc.

Application usage factor. Figure 1 indicates that people use different mobile applications when engaging different situations. For example, Paul uses a note application to record some important events or meeting summaries. While on the other hand, utilizes *Nike Plus* to record sports-related information when he is exercising.

Motion factor. In recent years, more people carry wearable device when they are exercising to record exercise progress and share them into social media. This trend motivates several researches to collect and analyze data of motion sensors which embedded in wearable device to distinguish some motion type activities. Based on those researches on the past, we observe that motion type activity has a special traits compared to other activities. Therefore, in addition to aforementioned features, we further adopt built in accelerometer in smart phone to discriminate motion type activity (e.g., sporting) from the others.

However, we encounter two challenges in using spatial data: 1) locational functionality, 2) discovering activity region. First, we observe that one location might provide different functionalities for each person. For instance, restaurant is a place for customer to eat but is a working place for the employee. Thus, we should personalize location feature for each individual. Second, in practice, we find that several POIs could refer to the same location. For example, movie theater and apparel shops are located in different parts of a shopping mall. Therefore, clustering nearby locations into a geographical region is necessary.

In this paper, we propose an activity inference model that integrate various features, such as: time, location, application usage, and accelerometer which can be collected from smart phone, to identify human activities. We separate our model into three components. (i) Motion feature. We extract and combine three axes accelerometer readings to form a synthesized magnitude[1]. (ii) Software feature. We discover top-k discriminative application list for each activity from each individual application usage log. (iii) Spatial-Temporal feature. First, we cluster all GPS points into several frequent regions. For each region, we analyze the frequency of activities and extract the temporal feature to form a distribution matrix. Furthermore, in case the user travels to a location where he has not been before, we take advantage of public opinion from the external dataset (adapted from [7]) to obtain general activity distributions for this region. Finally, we integrate these features to build activity inference model.

The main contribution of this paper are :

- We proposed an activity inference framework (3D-AIM) which consists of three components: Motion, Application, and Spatial-Temporal. Additionally, we utilized outsourced dataset to assist users' activity inference process.
- We proposed a novel algorithm that takes advantage of application usage for inferring activity. To the best of our knowledge, this is the precursory work that uses software concept to infer user's activities. Even by solely using this feature, the accuracy is still over 83%.
- We conduct comprehensive experiments on two real datasets. According to the results, the average accuracy of our activity inference model can reach 91%.

The rest of this paper is organized as follows: Section II presents several state-of-the-art related works on activity inference. Section III formally defines the terms used in this paper and describes the overview of proposed 3D-AIM framework. In Section IV, we introduce features in our model and classification method for activity inference model. Section V displays the comprehensive experiment results. We discuss various limitations and challenges in Section VI. We summarize the conclusions of our works and contributions in Section VII.

II. Related Work

We categorized existing methods into several groups: sensor-based approach, location-based approach, and hybrid approach.

A. Sensor-based Approach

Many researches collected environmental data from hardware sensors and trained an inference model to classify several activities or actions. For instance, Hung et al. [8] classifies some socially relevant actions by solely using a wearable device, which is attached on the body, to determine whether a person is speaking, laughing, gesturing, drinking, or walking. They create several different classifiers for each action to distinguish between the action and its absence. Subsequently, they trained two different Hidden Markov Models (HMMs) for each action. But their model only worked in dense crowded social gatherings, i.e., proximity between people is small. Furthermore, only face-to-face conversation can be recognized. Differently, the authors in [9] propose an approach to recognize daily routines, i.e., commuting, work, lunch, and dinner from wearable sensors. First, they collect and compute several features for supporting inference, including mean and variance of the 3D-acceleration signals from wrist and pocket, plus time-of-day information. Second, they utilize probabilistic topic models [10], [11] to automatically discover activity patterns from sensor data and to recognize daily routines by using probabilistic combination of activity patterns. Then, they adopt Naive Bayesian as classifier. However, they don't consider user's location information as a feature. Yet, we believe that the daily routines which they intend to recognize could

[1]Synthesized magnitude $= \sqrt{A_x^2 + A_y^2 + A_z^2}$

be identified well by utilizing temporal and location feature instead of discovering many activity patterns (e.g., walking freely, sitting, etc.). Moreover, they conduct experiments on only one user, therefore, performance of their model might not be objective enough. Differently, the authors in [12] only use phone-based accelerometers for physical activity recognition. They collected tri-axial readings of accelerometer and trained a predictive model to identify walking, jogging, climbing stairs, sitting, and standing. But their model can only classify motion type activities, inferring physical activities, instead of inferring high-level human activities (e.g., working, dining, etc.), which are more complex.

B. Location-based Approach

According to the example showed in Figure 1, we know that human's daily activities are highly related to where they are. Therefore, many researches have studied on inferring users' behavior by exposing the functionality or category of locations.

The authors in [13] study a location prediction in LBSNs. They decompose this problem into two components: 1) Predict the category of user's activity through mixed HMM which models user's movement pattern, temporal pattern, and dependency between check-in activities, and 2) Predict the locations that are more likely to be visited by utilizing category distributions which is extracted from the previous step. But the problem they want to solve is essentially different from ours. They predict a location where a user would visit in the future based on movement pattern history, while we utilize present information to infer user's current activity.

Differently, Zheng et al. in [14] propose CLAR model that support two modes of classification: 1) discovering interesting location based on queried activity, 2) ascertaining activities based on queried location (by using GPS data and users' comment which are annotated on the location). In other words, they mainly aim to solve two problems, one is location recommendation by given some activity queries, and the other is activity recommendation by given some location queries. For these purposes, they extract location features and activity-activity correlations from mobile data history, and users' comment from the Internet to form collaborative location-activity matrix for activity inference. But there are some drawbacks in their system. First, they don't consider user's personal preference. Different users at the same place would get the same recommendation from their model. Yet we argue that one particular place might have different meaning for different user. Second, their model could only perform recommendation if the queried location is in their database.

C. Hybrid Software and Hardware Approach

The authors in [15] combine software and hardware to form a low resource consumption model to classify different transportation modes: still, walking, running, biking, and on a vehicle. They collect movement data by using motion sensors' data on wearable device or mobile phone. The features that they use are: 1) average, 2) standard deviation, 3) highest frequency, 4) ratio between highest and second highest frequency, for all Accelerometer's, Gyroscope's, and Magnetometer's magnitude (i.e., $\sqrt{A_x^2 + A_y^2 + A_z^2}$), except for Magnetometer that doesn't use (1). Further, they select SVM as their classifier and set up optimal kernel and parameter in order to reduce resource consumption. In addition, they propose a voting scheme to reduce the error from the classifier when shifting between different transportation modes. Although their proposed model can reach high accuracy at low resource consumption, but they only focus on identifying the transportation modes and have no concern on differentiating transportation with other activities.

All in all, there are some drawbacks on aforementioned existing works. **Hardware and resources**. Here, we only take advantage of accelerometer, GPS, and application usage to infer users' activity, but some studies adopt more sensors for inferring user's action or behavior that leads to high power consumption. Furthermore, unlike some studies which use a particular device which is specifically made for research purposes, we utilize a commercial device, embedded sensors on mobile phone, that is widely available. **User preference**. Most of the related works' models don't take personal factor into consideration, yet we argue that every user might have different habits, thus we should design a personal activity inference model rather than a general one. **Inference restriction**. Several literatures stated above have some limitations, such as: proximity and data availability. Therefore, we aim to develop a model that not only has no setting limitation but also can infer user's activity without data availability requirements.

III. OVERVIEW OF FRAMEWORK

In this section, we define some terminologies and introduce 3D-AIM framework in a high level concept.

A. Preliminary

First, we clarify some terminologies used in this paper as follow:

Definition 1. Accelerometer Reading: Accelerometer reading $Accel$ is a time-stamped reading: $Accel = \langle x, y, z, t_i \rangle$, where x, y, and z are 3-axis value of accelerometer with a timestamp t_i that $t_i < t_{i+1}$.

Definition 2. GPS Points: GPS points GPS is a set of two-dimensional coordinates with timestamp: $GPS = \langle p_0, p_1, ..., p_n \rangle$, where n is the number of GPS points, $p_i = (lon_i, lat_i, t_i)$, lon_i and lat_i is the longitude and latitude of a location, and t_i is a timestamp that $t_i < t_{i+1}$. Figure 2a shows an example of a trajectory which consists of 7 GPS points.

Definition 3. Stay Point: A stay point SP is a geographical region where a user stayed over a time threshold T within a geographical distance D. We denote $SP = \langle lon, lat \rangle$ where lon and lat represent the longitude and latitude of the centroid of a set of consecutive GPS. Figure 2a shows an example of a stay point which consist of 4 GPS points (i.e., p3, p4, p5, p6) and red dot shows the centroid of the stay point.

Definition 4. Frequent Region Set: Let Mpt be minimum visit frequency, then frequent region set FRS is a set of geographical areas within a radius R where user's number of visit in the stay points $> Mpt$. We define $FRS = \langle fr_0, fr_1, ..., fr_n \rangle$ where n is the number of frequent regions , $fr_i = (lon_i, lat_i, SPs)$, lon_i and lat_i represent the longitude and latitude of the centroid of all SPs in fr_i. In Figure 2b, we demonstrate two frequent regions (these two frequent regions form a frequent region set for a user) which consist of several stay points.

Definition 5. List of Application Package Name: User list of application package name $LAPN$ indicates a set of applications which are used when user performs a particular activity. Formally, we define $LAPN = \langle apn_0, apn_1, ..., apn_n \rangle$, $\forall apn \in LAPN : apn_i \neq apn_j$ where $i \neq j$ and n is the number of applications.

Definition 6. Activity Log: There are six predefined activities Act in this paper: Working, Dining, Shopping, Transportation, Entertainment, and Sporting. For each activity Act, there exists activity logs $ActLog$, which list both hardware and software records from smart phone. We define as: $ActLog = \langle r_0, r_1, ..., r_n, Act \rangle$, where n is the number of records, $r_i = (Accel, GPS, LAPN, t_i)$, t_i is a timestamp which $t_i < t_{i+1}$.

B. Framework

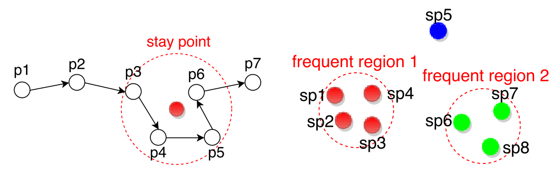
Figure 3 shows the overview of 3D-AIM framework which consists of two components: training stage and classification stage. In the training stage, we extract four features: 1) time, including time of the day and day of the week, 2) *GPS* readings, 3) *accelerometer reading*, 4) *list of application package* from $ActLog$ as the input. In addition, we also have activity label as the input for training stage. Subsequently, we put $ActLog$ records into all motion part, software part, and spatial-temporal part, and integrate training results from these three parts to build a personal activity inference model in the hybrid part. In the classification stage, we collect aforesaid four features for a period of time, then we have a two-step process for classifying activities: (1) We utilize a threshold model which is based on motion factor to identify motion type activity (e.g., sporting) first. (2) If threshold model couldn't recognize whether a activity is motion type activity or not, we combine all factors to further infer activities. In other words, step (1) is solely used to identify motion type activity, and step (2) is used to classify all activities. Thus, if we know that an unknown activity is a motion type activity at step (1), we won't perform step (2). The benefit of this two-step process is that it allows us to use less resources for identifying motion type activity. Furthermore, if a user visits a location where he has never been before, we also utilize public opinion from Foursquare to solve this problem.

IV. Methodology

In this section, we introduce proposed classification methods and explain how to transform raw input data into the features that we could utilize directly to train our activity inference model.

A. Fuzzy Activity Inference

First of all, we consider that any activity could be performed anytime anywhere. Here, we adopt the concept of fuzzy



(a) (b)

Fig. 2: An example of GPS, stay point and frequent region.

inference process [16] to infer user's activity, because the characteristics of fuzzy inference process are very appropriate for the problem we want to solve. The process demonstrates an approximation result (many-valued logic in degree) rather than an exact result (binary logic) which is generated by traditional classifiers.

Fuzzy inference process consists of three parts: fuzzification, rule evaluation, and defuzzification. We divided 3D-AIM framework into three inference parts: motion part, software part, and spatial-temporal part. They are relevant to fuzzification and rule evaluation. We translate each input, i.e., accelerometer readings, application package name, and spatial-temporal data into activity probability distributions, and calculate three scores from each part for each activity. However, generated probability matrices for each part are different, resulting in different inference result. Therefore, we need hybrid part (corresponds to defuzzification) to integrate three relative results for each activity, and then select the activity with the highest probability as the final inference result. We present each part in the following subsections separately.

B. Motion Part

Since people might carry wearable or mobile device in any orientation, it would be better to analyze all three axes accelerometer readings altogether to measure overall motion in any orientation. Therefore, we utilize magnitude of accelerometer [15] that could combine all three axes acceleration values. Furthermore, acceleration value might vary a lot within milliseconds interval, therefore we need to group acceleration readings within one second interval to reduce the noise. The definition of average magnitude of acceleration value per second A_{mag} is:

$$A_{mag} = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (1)$$

First of all, we extract $Accel$ from collected $ActLog$. In our preliminary study, we evaluate four features, including (1) standard deviation of A_{mag}, (2) highest value of A_{mag}, (3) lowest value of A_{mag}, and (4) average of A_{mag}. But later, standard deviation of A_{mag} is selected because of its ability to distinguish motion type activity perfectly. Further, motion part is implemented by using two strategies: threshold-based model and distribution matrix.

In the threshold-based model, we build a precision-recall graph to determine the optimal threshold for discriminating motion type activity from the others. If standard deviation of

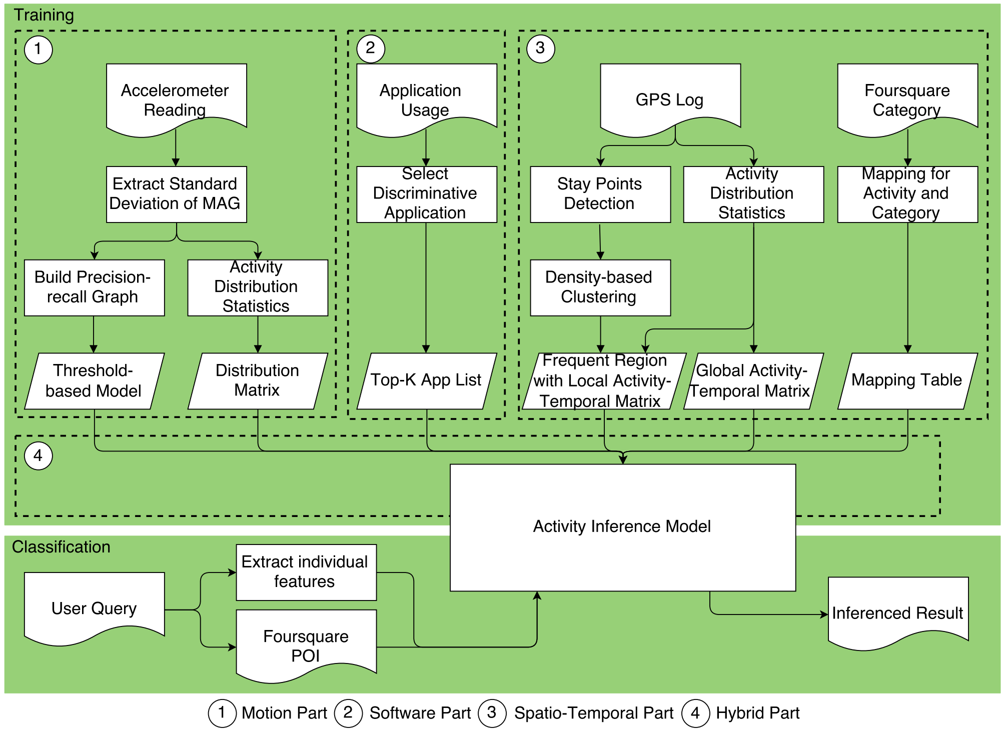


Fig. 3: An overview framework of 3D-AIM.

TABLE I: An example of distribution matrix of standard deviation of A_{mag}.

Standard Deviation	Working	Dining	Transportation	Sporting	Shopping	Entertainment
0.00	0.50	0.30	0.00	0.00	0.20	0.00
0.10	0.45	0.25	0.15	0.15	0.00	0.00
0.20	0.40	0.20	0.15	0.10	0.10	0.50
...				...		

A_{mag} during a period of time is over than a particular threshold, then threshold-based model will directly recognize the activity as motion type activity. Figure 4 shows the precision-recall graph of standard deviation of A_{mag} in our dataset. Each point on the graph separately represents the standard deviation of A_{mag} that ranges from 0 to 2.4. The precision-recall graph displays the percent of recall and precision against the standard deviation of A_{mag}. Here, threshold-based model could distinguish motion type activity perfectly if the standard deviation of A_{mag} is higher than 2.3. This phenomenon represents real-world situation well, as motion type activity (e.g., sporting) requires more extreme movements than other activities (e.g., working, eating, sleeping, commuting, etc). Further, percentage of recall depicts how much data is covered. The advantage of using threshold-based model is that we could classify motion-type activity directly because of its nature, without further evaluating other factors. This approach allows us to reduce resource consumption (e.g., computing, memory, energy, etc) without sacrificing accuracy of the result. However, if standard deviation of A_{mag} is under 2.3, the threshold-based model can't classify any activity. Therefore, we have another approach to handle this situation. To further evaluate our decision on selecting the threshold (i.e., 2.3), we also evaluate the value by using decision tree in R [17] and Weka [18] to determine the optimal threshold for classifying sporting and other activities. Finally, both R and Weka also consider 2.31 as the best threshold, which is similar with our threshold result.

In the distribution matrix, we build and normalize a personalized standard deviation distribution matrix of A_{mag} $SDDM$ that contains the distribution of standard deviation of A_{mag} for six activities that we have mentioned in Section III. The definition of $SDDM$ is presented as follow:

$$SDDM_{i,j} = \frac{f_{i,j}}{\sum_{j=1}^{M} f_{i,j}}, \forall j \in Act. \tag{2}$$

where $f_{i,j}$ is frequency of activity j performed at standard deviation of A_{mag} i. Table I exhibits an example of $SDDM$ for motion part.

In summary, if standard deviation of A_{mag} is over 2.3, then motion type activity is inferred directly. On the contrary, if standard deviation of A_{mag} is under 2.3, our activity inference model is able to classify activities by using $SDDM$. Yet, in this case, all three aforementioned factors need to be utilized together to obtain the final inference result. In the other words,

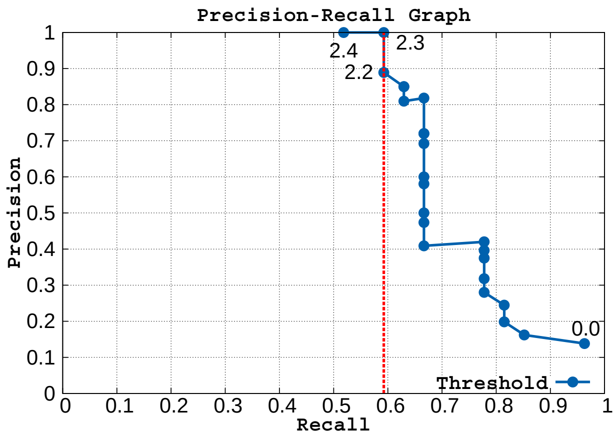


Fig. 4: Performance of different threshold.

by using distribution matrix, we couldn't obtain the benefit of using less resources.

C. Software Part

First of all, we extract application package names from user's $ActLog$ to form a $LAPN$ for each activity. As we have six activities, thus we also have six $LAPN$ that correspond to each activity that we have defined in Section III. For each $LAPN$, we construct discriminative applications pattern to identify each respective activity. To measure the capability of each app, we calculate the score of each app based on its rank. Naively, utilizing usage frequency to rank activity discrimination ability of each app is an intuitive approach. However, using this method to rank apps for identifying activity would be useless when user always use the same apps whatever he does. For example, if a user uses Facebook much more than Gmail when dining, Facebook would be a better application pattern for identifying dining. But, if a user uses Facebook whatever he does, this app would be useless because it couldn't classify any activity. Therefore, we propose Entropy-Frequency to rank apps because this method considers not only usage frequency within each activity but also entropy between activities for each app. In addition, we also adopt Term Frequency-Inverse Document Frequency, which is often used in text mining to reflect how important a term is, to rank apps. This method is very appropriate for solving our problem because apps and $LAPN$s can be entirely mapped into terms and documents respectively. All in all, we propose three methods to calculate score for each application, including Frequency (F), Entropy-Frequency (EF), and Term Frequency-Inverse Document Frequency (TF-IDF) [19], which are defined as (3), (4), and (5) respectively:

$$F(app_{i,j}) = \frac{f_{app_{i,j}}}{\sum_{i=1}^{N} f_{app_{i,j}}} \tag{3}$$

$$EF(app_{i,j}) = [1 - (-\sum_{j=1}^{M} p_{i,j} \log p_{i,j})] \cdot F(app_{i,j}) \tag{4}$$

$$TF - IDF(app_{i,j}) = F(app_{i,j}) \cdot \log \frac{M}{\mid app_i : app_{i,j} \mid} \tag{5}$$

where i is each application, j is a given activity, $f_{app_{i,j}}$ and $p_{i,j}$ represent usage frequency and used probability of the application i in the activity j. However, if we take all used apps to construct personal application lists for identifying activity, it would make not only model size large but also classification time long. But from our observations, not each app can be useful, i.e., the apps with high entropy. Therefore, after ranking activity discrimination ability of apps, we should select top-k applications to form an identification application list for each activity. In addition, in our preliminary study, we have extracted both app name and app category (according to Google Play) to form a list for each activity. Yet, the performance of using app category is not satisfying enough because every category appear in every activity, making its inference power low. Here, we propose two scoring method: general scoring and weighted scoring. In *general scoring*, we don't consider the rank of the application. If an application matches with top-k applications for certain activity, then that activity would get one point. On the other hand, in *weighted scoring*, we consider the rank of the application in top-k applications. The score decreases progressively with the rank. For instance, the first application in top-4 application list would get 4 points, and the last application would get 1 point.

By using the example in Figure 5, we could select top-4 apps (k=4) for each activity (i.e., working, sporting, dining). User might use many applications when performing activity, but here, we assume that user is doing a certain activity and using only five applications: Notepad, BBC News, Gmail, Instagram, and Yahoo Weather on smart phone. We would identify user's activity by observing his application usage pattern. Calculating score for each activity is shown in Table II. In general scoring, the score for working, sporting, and dining are: 2, 1, and 3 respectively. While in weighted scoring, the working score is 7, sporting score is 1, and dining score is 6. After calculating scores, we can find that inferred activity of general scoring is dining, but inferred activity of weighted scoring is working.

TABLE II: An example of calculating for software part.

Activity	General Scoring	Weighted Scoring
Working	33%	**50%** (4+3)
Sporting	17%	7% (1)
Dining	**50%**	43% (3+2+1)

D. Spatial-Temporal Part

By extracting individual geographical feature, we could find out the regions where user frequently visited and what kinds of activities that user perform in these regions, i.e., the functionality of location for each individual. Likewise, by extracting

TABLE III: An example of $LATDM$ and $GATDM$.

(a) Local activity-timestamp distribution matrix of frequent region fr_i.

Time Slot	Working	Dining	Transportation	Sporting	Shopping	Entertainment
...				...		
8-9	0.40	0.20	0.00	0.20	0.10	0.10
...				...		
15-16	0.70	0.00	0.00	0.00	0.10	0.20
16-17	0.00	0.00	0.00	0.00	0.00	0.00
...				...		

(b) Local activity-timestamp distribution matrix of frequent region fr_j.

Time Slot	Working	Dining	Transportation	Sporting	Shopping	Entertainment
...				...		
15-16	0.20	0.00	0.00	0.40	0.00	0.40
16-17	0.40	0.60	0.00	0.00	0.00	0.00
...				...		

(c) Global activity-timestamp distribution matrix.

Time Slot	Working	Dining	Transportation	Sporting	Shopping	Entertainment
...				...		
8-9	0.40	0.20	0.00	0.20	0.10	0.10
...				...		
15-16	0.45	0.00	0.00	0.20	0.05	0.30
16-17	0.40	0.60	0.00	0.00	0.00	0.00
...				...		

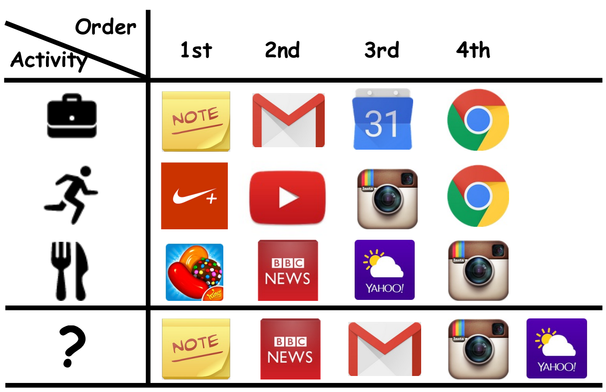


Fig. 5: An example of user's application list when performing certain activities.

individual temporal feature, we could discover user's daily life routines which is modelled by using timestamp. From Figure 1, we know that user is in the office by extracting GPS data. Yet, we observe that user might perform two different activities in the region (office). Hence, we combine both spatial and temporal data together as one factor to infer user's activity.

First, we collect user's raw GPS data from $ActLog$. Subsequently, we detect stay points from these GPS by adopting the algorithm in [20]. As compared with raw GPS points, stay points are more meaningful because stay points have longer time duration than GPS points, therefore it could represent richer semantic meaning (i.e., activity which could be performed in the location), for example, the place we work, the restaurant we visit, etc. However, in practice, we find that several stay points might be referring to the same location. For example, movie theater and apparel stores are located in different parts of a shopping mall. Therefore, we need to further cluster nearby stay points into a geographical region to extract richer meaning of the location. Here, we utilize a density-based approach *OPTICS* [21], rather than grid-based approach, to cluster stay points into personal FRS, because generally frequent regions tend to have irregular shapes. Each gps point $p_i = \langle lon_i, lat_i, t_i, act \rangle$ in GPS also contains timestamp and activity information. However, we argue that in the same time slot, user could perform more than one activities within a frequent region, therefore we can't directly infer user's activity. Based on this observation, we build and normalize a local historical activity-timestamp distribution matrix $LATDM$ that contains frequency distributions of six activities that we have mentioned in Section III on each frequent region. We divide one day into 24 time slots according to the 24 hours of the day. The definition of $LATDM$ is presented as follow:

$$LATDM_{i,j} = \frac{f_{i,j}}{\sum_{j=1}^{M} f_{i,j}}, \forall j \in Act. \tag{6}$$

where $f_{i,j}$ is the frequency of activity j performed at time slot i.

Table IIIa and Table IIIb represent $LATDM$ of frequent region fr_i and fr_j respectively. Using Table IIIa as an example, the user has 210 records of working, 30 records of shopping, and 60 records of entertainment from 3 to 4 p.m. (including weekday and weekend) on frequent region fr_i, then we normalize $LATDM$ and obtain the activity probability at this time slot: working is 70%, sporting is 10%, entertainment is 20%, and others are 0% because other activities aren't performed at this time period.

Moreover, there is a possibility that user perform an activity at a certain period of time even though he has never performed any activity before at this time slot in a frequent region. Using Table IIIa and Table IIIb as an example, the user might have an afternoon tea at frequent region fr_j at 4:30 p.m. (probability of dining is 60% from Table IIIb), but within frequent region fr_i, there is no any record of user performs any activity at this time period. In other words, if the user performs an activity at 4:30 p.m. on fr_i, we couldn't infer what he does by solely using $LATDM$ of fr_i. Thus, in addition to $LATDM$ of frequent region fr_i and fr_j, we also build a global activity-timestamp distribution matrix $GATDM$ which combines all activity-timestamp distributions together from overall $LATDM$ for obtaining inference hints through user's habits on other frequent regions. Therefore, we can infer that user might perform dining from $GATDM$. The definition of $GATDM$ is presented as follow:

$$GATDM_{i,j} = \frac{\sum_{k=1}^{N} f_{i,j,k}}{\sum_{k=1}^{N}\sum_{j=1}^{M} f_{i,j,k}}, \forall j \in Act. \tag{7}$$

where $f_{i,j,k}$ is the frequency of activity j performed at time slot i in frequent region k, and N is the number of frequent regions. $GATDM$ is employed iff respective frequent-region couldn't provide the probability of performed activity, otherwise, we use $LATDM$ to provide the probability of the activity. As shown in Table IIIc, the structure of $GATDM$ is similar with $LATDM$, it consists of time slots and probability distribution of each activity in each time slot. In summary, if the user performs a certain activity in frequent region fr_i at a certain period of time for the first time (i.e., he hasn't never performed any activity at this time slot), probabilities of all activities at this time slot in $LATDM$ of frequent region fr_i are 0%, (e.g., probabilities during 16-17 in Table IIIa), however, we still can infer what the user performs by selecting activity with the highest probability at this time slot from $GATDM$.

However, we observe that people tend to visit a location where he has never been before (e.g., when having a vacation trip or conference). In this case, the frequent region hasn't been discovered, implying that no $LATDM$ exist in this region, therefore, we can't extract any information from the location, as it is not available in the database. Yet we might get some information from (1) $GATDM$ and (2) the location by referring to some public opinions from the Internet. Then we combine these two factors together to handle with this situation.

Although users might visit a location where he has never been before, they still could perform some daily routines on that location. For example, if a user first visits a location at noon, he might have a lunch there. Because people often have a lunch at noon wherever they are. Therefore, we should take activity probability from $GATDM$ into consideration because we believe that $GATDM$ could reflect individual habits in daily life.

Besides $GATDM$, we also refer to some public opinions from the Internet. There are many websites providing POI (point of interest) category query, such as: Foursquare, Gowalla, WikiMapia, etc. In this paper, we adapt a work from [7] that uses Foursquare as the reference for POI categorization. Moreover, Foursquare has a vast amount of users around the world and exhibits a large number of POI categories. Their approach on using public opinion are described as follows. First, if the user visits a location where he has never been before, they extract the longitude and latitude of the location and search the category of POI within a specific distance D on Foursquare. In other words, they obtain a set of category of POI in the circular region that is formed from the position where user has never been to the specific distance D (as illustrated in Figure 6). Second, they count the frequency of each category to get category distribution and normalize the distribution of POI's categories by utilizing TF-IDF to increase the weights of categories that are unique in the region.

Subsequently, they transform the POI categories in Foursquare into our predefined six activities. Table IV (adopted from [7]) presents the mapping table of location category of Foursquare and each activity. There are nine high level categories of POI in Foursquare, each category is mapped into one activity by using common sense. We find out that only a few places particularly provide exercising service in these high level categories. Therefore, there is not any high level category is mapped into sporting. Yet, here, we try to compromise this shortcoming by utilizing Motion factor (in section 4-2), which focuses on motion type activity, including sporting activity. Finally, by using the example presented in Figure 6, mapping table displayed in Table IV, we could get the public functionality distribution in the region where user has never been before: shopping is 40%, entertainment is 29%, and dining is 31%.

TABLE IV: Mapping table of location category and activity.

Location Category	Activity
Food	Dining
Shop & Service	Shopping
Travel & Transport	Transportation
College & University	Working
Professional & Other Places	Working
Residence	Entertainment
Nightlife Spot	Entertainment
Arts & Entertainment	Entertainment
Outdoors & Recreation	Entertainment

In summary, we integrate public opinions and activity probability of $GATDM$ to infer user's activity when he visits a location that he has never been before.

E. Hybrid Part

In this part, according to the final step of fuzzy inference process, we introduce the combination process of three probability distributions from the motion part, software part, and spatial-temporal part for each activity. We integrate three scores from each corresponding part for each activity through Equation 8:

$$\alpha \begin{bmatrix} ms.w \\ ms.d \\ ms.t \\ ms.e \\ ms.sp \\ ms.sh \end{bmatrix} + \beta \begin{bmatrix} ss.w \\ ss.d \\ ss.t \\ ss.e \\ ss.sp \\ ss.sh \end{bmatrix} + \gamma \begin{bmatrix} ls.w \\ ls.d \\ ls.t \\ ls.e \\ ls.sp \\ ls.sh \end{bmatrix} = \begin{bmatrix} p.w \\ p.d \\ p.t \\ p.e \\ p.sp \\ p.sh \end{bmatrix} \tag{8}$$

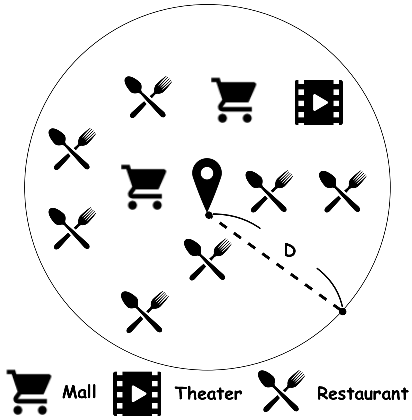


Fig. 6: Category distribution of POI in a region where user has never been before.

where ms represents motion score, ss represents software score, ls represents location score, and p represents activity probability.

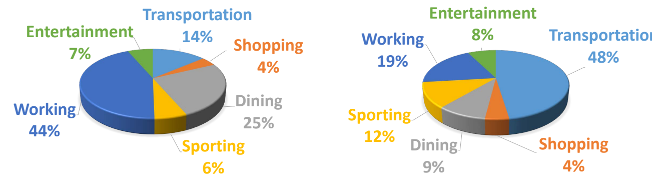
The probability of each activity is a real number between 0 and 1, where 0 means never performed and 1 means always performed. In addition, each feature might have different weight for each person. Thus, we set each coefficient α, β, and γ by rule of thumb for motion part, software part and spatial-temporal part so that our activity inference model gets personalized for each individual. Finally, we choose activity with the highest probability as the inferred activity.

V. Experiment

In this section, we conduct some comprehensive experiments on our activity inference model. We divide this section into three subsections: dataset description, parameter sensitivity analysis, and performance evaluations.

A. Dataset Description

We conduct comprehensive experiments on two real life datasets which are collected from 18 participants. The users utilize the activity logger on Android phone to record application usage, accelerometer readings, GPS points, time and manually label the activity. Our participants are required to record their start time and finish time of their activities records. There are six predefined activities in our experiments: $Working, Dining, Shopping, Transportation, Entertainment$, and $Sporting$. To be precise, dataset I has 18 participants from August 2013 to July 2015, which consists of 85,515 GPS points, while dataset II has 14 participants from May to August 2013. As dataset 2 doesn't have App information, therefore we only have two factor to be used in our model: motion and spatial-temporal factors. Yet, based on our observation on the dataset II, there are more people bring their smart phone when exercising (i.e., accelerometer value readings has more extreme value) on sporting activity. We divide the datasets into 80% training data and 20% testing data for each user. Figure 7 displays the activity distribution on our datasets.



(a) Dataset I

(b) Dataset II

Fig. 7: Activity distribution

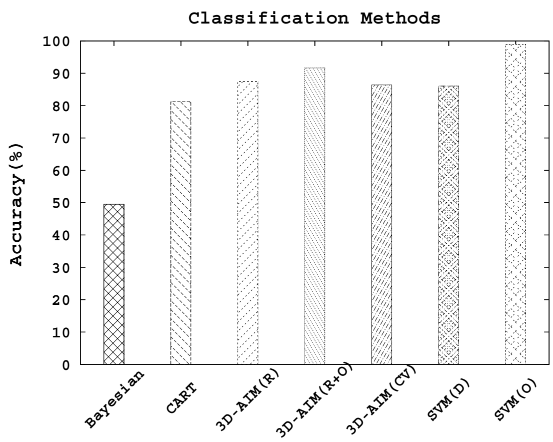
B. Performance Evaluation

In this section, we evaluate the performance for each part and overall system, including: 1) Overall Accuracy Evaluation, 2) Threshold-Based Model, 3) Inference Ability of Different Activity, 4) Inference Ability of Different Feature, and 5) The Resource Consumption.

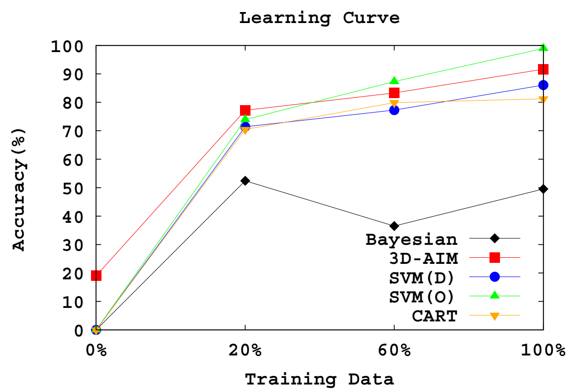
1) Overall Accuracy Evaluation: Average activity recognition accuracy performances between each model are shown in Figure 8a. Here, we compare our model with three other commonly used classification methods: Naive Bayesian, CART [22], and SVM. We implemented Naive Bayesian and CART using R programming language and adopt libSVM [23] for implementing SVM. In our model, we have also tested several environments to identify the capabilities of our proposed framework. We have 3 testing environments for our framework: Random (R), Optimized (O), and Cross validation (CV). We adopt 5-fold cross validation in this experiment. The kernel function used for libSVM is Radial Basis Function (RBF), and default penalty parameter C=1. Overall results show that 3D-AIM performs better than every methods but optimal libSVM. However, optimal libSVM requires longer time to train the model in order to obtain optimal penalty parameter.

The Figure 8b displays the learning curve for each classification method. Here, our proposed method outperforms other methods when training data is small. This indicates that our method only requires small amount of training data to be able to infer the activity, achieving about 75% accuracy by only using 20% of total training data. In addition, SVM isn't able to infer users' activity when there is no any training data at all, but our model still can infer activities by using public opinion with about 20% average accuracy.

2) Threshold-based Model: In this section, we utilize some statistics measures derived from confusion matrix to evaluate performance of threshold-based model, including precision, recall, accuracy, negative predictive value (NPV), whose results are presented in Figure 9. The precision reach 100% because we choose the threshold with the highest precision, therefore we don't have any false positive. While we could perfectly identify sporting by using the specified threshold, but we could only achieve 60% recall performance because some users might not carry their smart phone when they are sporting (e.g., swimming, soccer, basketball, etc.). This implies that some instances of sporting activities might be classified as other activity, hence, here, NPV value couldn't reach 100%. Finally, overall accuracy of threshold-based model can reach



(a) Average performance.



(b) Learning curve.

Fig. 8: Average performance and learning curve.

94%.

3) Inference Ability of Different Activity: Performances of each single activity are exhibited in Figure 10. Here, entertainment suffer the lowest score. One possible reason is because the total training data of entertainment data is relatively fewer than other activities and it often overlaps with activities like shopping and sporting, because the definition of entertainment is vague. In addition, entertainment activity is more difficult to be modelled over time or location constraint, because there are various possibilities of people doing entertainment. For example, it is challenging to differentiate between watching movie at home, and working at home, as they are both located at home and has less motion activity. Thus, inferring whether a user is doing entertainment is difficult than others. However, working and dining have high accuracy because most people usually regularly performs these activities in particular location and time. For example, people usually works in company in daytime and having lunch or dinner in the restaurant at noon or evening. Therefore, inferring whether a user is doing working or dining is intuitive by using spatial-temporal feature. Nevertheless, overall accuracy of all activities, except for entertainment are above 60%.

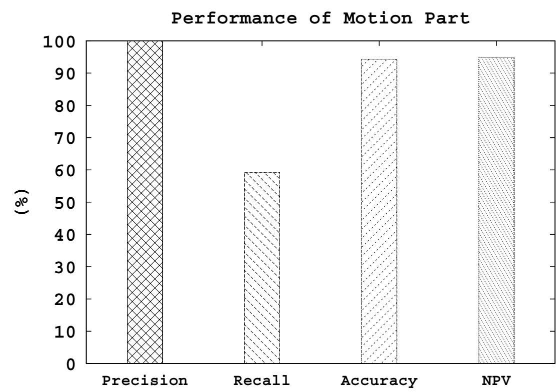


Fig. 9: Performance of threshold-based model.

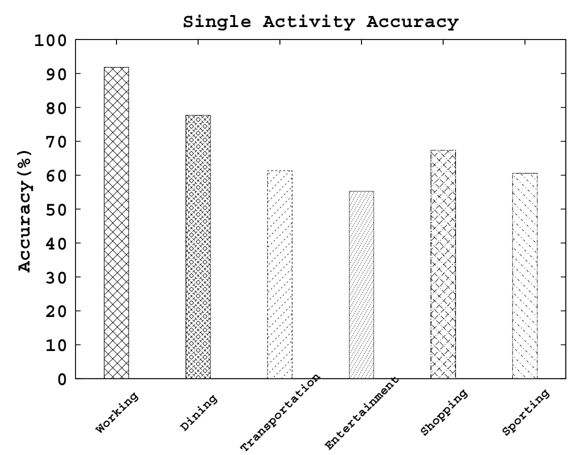


Fig. 10: Single activity accuracy.

4) Inference Ability of Different Feature: Performances of each combination of different features are exhibited in Figure 11. In this figure, M, S, and ST denote that we utilize motion, software, and spatial-temporal feature to infer activities separately. $M+S$ indicates that we combine motion and software feature together to infer activities (it's the same for $M+ST$ and $S+ST$). According to our hypothesis, classifying complex human activity (e.g., working, dining, shopping, etc.) by solely using accelerometer is difficult. Here, based on Figure 11, it is shown that the result is in accordance with our hypothesis (Without any support from other features, Motion part performs the worst). As expected, we get the highest performance when we use all features. However, we discover two issues in Figure 10: (1) performance of $M+S$ is worse than S and (2) inference ability of using all features is equivalent to $S + ST$. In other words, the impact given by motion part is low or even negative. One possible reason is that there is no significant difference in standard deviation of A_{mag} between all activities except for motion type activity. Therefore, motion part could identify motion type activity well but it couldn't classify other activities accurately. Yet, we still utilize motion part to infer activities because of its benefit (i.e., differentiating between motion type activity from others fast and efficiently).

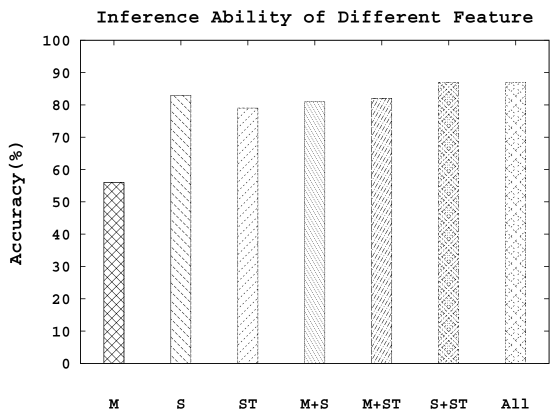
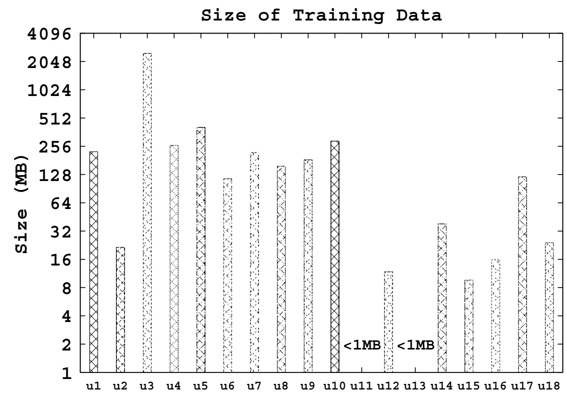


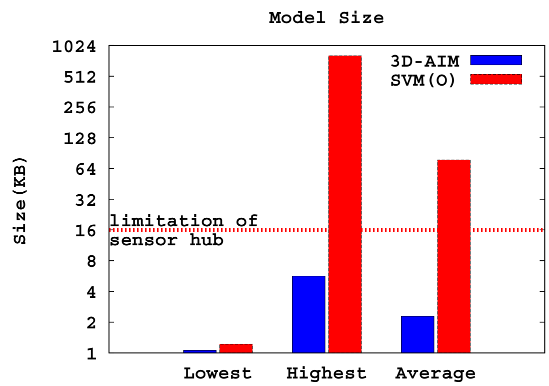
Fig. 11: Inference ability of different feature.

5) The Resource Consumption: In this section, we compare several resource consumptions with optimal SVM.

a) Average Execution Time: Average execution time of our model is better than optimal SVM, especially in training phase, as optimal SVM requires more time to find the optimal parameter (Figure 12). Therefore, in larger dataset, SVM suffers more time to train the data. Within our dataset, we can always perform the training and testing within 0.1 second which is acceptable for inferring activity in smart phone.



(a) Size of training data



(b) Model size

Fig. 13: Size of training data and inference model.

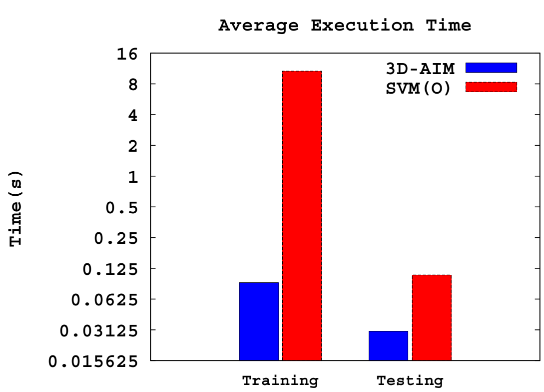


Fig. 12: Average execution time.

b) Model Size: Model size is an important issue for classifying human activities on mobile phone. Model size needs to be small enough so that the inference model can be built in smart phone. While building the model in smart phone is good, yet, we argue that the power consumption in smart phone CPU is still too high for all-day sensing. Thus, here, we also consider on using a micro controller (e.g., Sensor Hub) which specializes in sensing data and uses less power. In general, Sensor Hub uses smaller memory (i.e., 16 KB [24]). Therefore, the model size should be smaller than this limitation. Figure 13a shows the size of training data for each user and Figure 13b displays the difference of model size between SVM and ours. According to the result, space requirement of our model are smaller than optimal SVM no matter how large the data is (i.e., up to 2GB in our experimental data). Moreover, our model could adhere the requirement of Sensor Hub's memory constraint.

C. Parameter Sensitivity Analysis

In this section, we evaluate the impact of various parameters in our experiments, including: stay point detection parameters, frequent region extraction parameters, and top-k applications selection parameter.

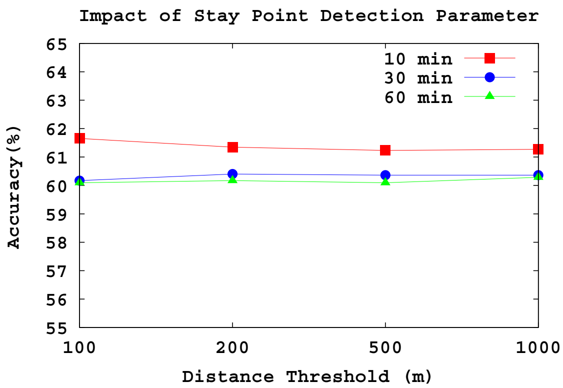


Fig. 14: Impact of stay point parameters.

1) Impact of Parameters in Stay Point Detection: In this experiment, we compare the performance of each assorted combinations of time threshold T and distance threshold D for each user in order to find the optimal parameters for detecting stay points. There is no significant difference (i.e., only two percents difference) between each setting (Figure 14). Nevertheless, we determine 10 minutes and 100 meters as the time threshold T and distance threshold D respectively.

2) Impact of Parameters in Frequent Region Extraction: In this experiment, we compare the activity recognition performance of different settings of minimum points Mpt and radius R for each user to find the optimal parameters for extracting frequent regions. In all cases, accuracies of activity inference descend along with the increasing of the minimum points and radius (Figure 15). Yet, parameter radius has bigger contribution to such performance degradation. All in all, the most optimal parameters for frequent region extraction are: Mpt = 2 and radius = 100m.

3) Impact of Parameter in Top-k Application List: In this experiment, we aim to discover the optimal k value of the application list. Figure 16 shows that k=30 is the best parameter for inferring activity through applications pattern. We have several observations from the dataset. First, people tend to use more than one application when they are performing an activity. Second, there are two types of applications: single purpose applications and general purpose applications, based on their activity coverage (i.e, single purpose applications cover a small number of activities, e.g., Nike Plus, and general purpose applications cover a lot activities, e.g., Facebook). Third, inference ability of single purpose applications is better than general purpose applications because single purpose

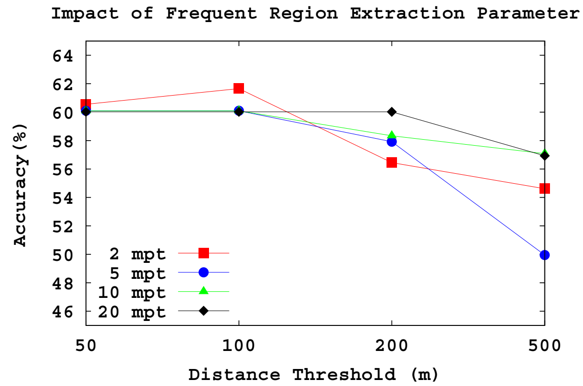


Fig. 15: Impact of OPTICS parameters.

applications have low entropy on deciding the activity. Fourth, only a small number of applications are categorized as single purpose applications. Fifth, people usually use general purpose applications, but such applications couldn't help much on depicting user's activity. Based on these observations, we need many applications to accurately classify different activity. Therefore, we determine to select top 30 applications to form a discriminative list.

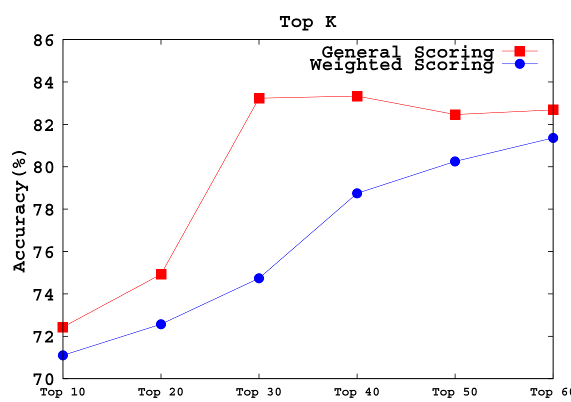


Fig. 16: Impact of top k for application list.

VI. Discussion

In the following subsections, we briefly address the challenges and limitations of our proposed methods.

A. User Behavior Sparseness

As recording process need users consent, because of the need to label the activity manually, therefore one data sparseness problem emerges - Variation: some users only have one or two types of activities. Although we have filtered user who only has one activity in his $ActLog$, yet the minimum number of activity types is still low (i.e., 2 activities). If the number of activities is low, then it would not reflect real world situation well.

B. Recording Problem

Confusion of data label might exist because of inaccurate labelling. For example, a user drives a car to a restaurant for dining, but he labels "dining" on the whole process. In order to make it correct, these records should be split into two activities information: transportation and dining. As definition of each activity is different for each person, logged activity records might suffer some confusions due to variation of user's viewpoint. For instance, entertainment often overlaps with activities like shopping and sporting because entertainment is self-defined type activity, i.e., different types of activities could be treated as entertainment, using one's perspective. Based on these characteristics, in some cases, 3D-AIM's performance couldn't reach its maximum, due to its dependency on trained labels.

C. Threshold-based Model Gain

In our 3D-AIM framework, we utilize threshold-based model to discriminate motion-type activity from other activities. We evaluate performance of threshold-based model only on dataset II because few users carry their smart phone when they exercise in dataset I (i.e., not many instances of sporting activities in dataset I could be classified from threshold model). According to Figure 7 and Figure 9, ratio of sporting activity is 12% of dataset II, but only 60% of sporting activity are inferred correctly, i.e., there is 7.2% of data that could use threshold model directly. However, we believe that it's helpful for saving resource when user is sporting, as sporting activity has distinguishing feature on accelerometer readings. Even though we couldn't measure the benefit of our approach quantitatively, yet if sporting activity inference process could be simplified by using one low-power sensor (i.e., accelerometer), then one could reduce overall inference power consumption by turning off some high-power sensors (e.g., GPS). Higher ratio of sporting activity leads to higher efficiency over power saving from our proposed threshold-based model. All in all, by applying two-step classification by using motion feature, we might gain some benefits without having any losses, as calculated motion part's data still could be used if threshold-based model fails to distinguish motion-type activity.

D. Application Usage Analysis

Based on our observation in the dataset, we discover that build-in apps (e.g., music player) are dominating in number and perform well in most cases, i.e., there is less public app (which is available from official market, e.g., $GooglePlay$) which could perform an accurate activity inference for all users. Therefore, personalization of application usage is crucial, i.e., inference model couldn't provide good performance by using single model for all users. In addition, we found that by solely using application usage feature, working perform the best of all (over 10% higher compared to other activities). One intuitive reason is that training data of working is more than other activities. Another possible reason is that people tend to use particular apps or even use fewer apps (compare to other activities) when they are working. On the contrary, inference accuracy of entertainment and dining is the lowest because users might have more time to use many apps on smart phone.

E. Public Opinion Gain and Limitation

Our 3D-AIM framework can reach 20% inference accuracy by using public opinion even without any training data. However, using public opinion is useless for overall performance when we have training data. As we mentioned in last section, we divide dataset into 80% training data and 20% testing data. Subsequently, we filter out some records that have no gps data. Within our testing data (all records have gps data after filtering), there are only 22% of the data that their locations are in users' frequent region. We further observe that 45% of the data which their locations aren't in users' frequent region are transportation activity. We argue that transportation activity couldn't form a frequent region, as transportation usually represents a movement of user over long distance within short period. Therefore, we utilize both personal (i.e., $GATDM$) and public dataset in synergy and only use public opinion as a supplementary data, when the user is not in his/her frequent region, as in many cases activity-timestamp distributions might have personal meanings toward users.

F. Features Preference

In our 3D-AIM framework, we utilize rule of thumb to determine different coefficients (i.e., α, β, and γ) for different features in hybrid part because each feature might have different weight for each person. In our study, we have proposed three approaches to determine coefficients: entropy of activity probability, inference ability of feature, and rule of thumb. In entropy of activity probability, we have the probability of each activity for each feature, the feature with lower entropy of activity probability gets higher weight. In inference ability of feature, the more powerful the inference ability of feature is, the higher the weight is. Yet, the performances of entropy of activity probability and inference ability of feature are not satisfying enough, they are both worse than rule of thumb. Therefore, we adopt rule of thumb to determine coefficients for different features.

VII. Conclusions

In this paper we proposed a framework for personal activity inference. Our approach consists of two components: training phase and classification phase. In training phase, we extract four features: time, GPS, accelerometer readings, and applications from activity log data and build a personal activity inference model by utilizing these information. In classification phase, we not only collect those four aforementioned features but also utilize public opinion from the Internet (i.e., Foursquare query) to infer the activity.

We also conduct comprehensive experiments on two real life datasets to evaluate the effectiveness of the proposed method. The experiment results show that our model can reach outstanding performance, in terms of the balance between accuracy, running time, and model size. Here, we take advantage of app usage patterns and accelerometer readings to support the inference process in case of having weak GPS signal (e.g., indoor). We also utilize public opinion from Foursquare so that our model could work even without training data at all.

REFERENCES

[1] W. A. Social. (2014, Nov.) Global digital statistics. [Online]. Available: http://wearesocial.net/tag/stats/

[2] Forrester. (2013, Jun.) Smartphone usage. [Online]. Available: http://www.iqmetrix.com/article/2013/06/forrester-finds-differences-tablet-smartphone-usage

[3] Google now. [Online]. Available: http://www.google.com/landing/now/

[4] Motorola assist. [Online]. Available: http://www.motorola.com/us/consolidated-apps-page/motorola-assist.html

[5] Yahoo aviate. [Online]. Available: http://aviate.yahoo.com/

[6] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2010.5560598

[7] X. W. Ruan, S. C. Lee, and W.-C. Peng, "Exploring location-related data on smart phones for activity inference." in *MDM (2)*, 2014, pp. 73–78.

[8] H. Hung, G. Englebienne, and J. Kools, "Classifying social actions with a single accelerometer," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13. New York, NY, USA: ACM, 2013, pp. 207–210. [Online]. Available: http://doi.acm.org/10.1145/2493432.2493513

[9] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, ser. UbiComp '08. New York, NY, USA: ACM, 2008, pp. 10–19. [Online]. Available: http://doi.acm.org/10.1145/1409635.1409638

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[11] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1-2, pp. 177–196, Jan. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1007617005950

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011. [Online]. Available: http://doi.acm.org/10.1145/1964897.1964918

[13] H. Cheng, J. Ye, and Z. Zhu, "What's your next move: User activity prediction in location-based social networks," in *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, 2013, pp. 171–179. [Online]. Available: http://dx.doi.org/10.1137/1.9781611972832.19

[14] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with gps history data," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 1029–1038. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772795

[15] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1429–1440, Aug. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2733004.2733015

[16] O. Valenzuela, I. Rojas, and F. Rojas, "Statistical analysis of the different operator involved in the fuzzy inference process," in *Fuzzy Logic and Applications*, ser. Lecture Notes in Computer Science, V. Di Ges, F. Masulli, and A. Petrosino, Eds. Springer Berlin Heidelberg, 2006, vol. 2955, pp. 63–71. [Online]. Available: http://dx.doi.org/10.1007/10983652_9

[17] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[19] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[20] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '08. New York, NY, USA: ACM, 2008, pp. 34:1–34:10. [Online]. Available: http://doi.acm.org/10.1145/1463434.1463477

[21] M. Ankerst, M. M. Breunig, H. peter Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure." ACM Press, 1999, pp. 49–60.

[22] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*, ser. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. [Online]. Available: https://books.google.ca/books?id=JwQx-WOmSyQC

[23] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199

[24] Fxlc95000cl: Intelligent motion-sensing platform / freescale. [Online]. Available: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FXLC95000CL