Exploiting Joint Wifi/Bluetooth Trace to Predict People Movement

Long Vu, Quang Do, Klara Nahrstedt
Department of Computer Science, University of Illinois
Email:{longvu2,quangdo2,klara}@illinois.edu

***Abstract*—It is well known that the daily movement of people exhibits a high degree of repetition in which people usually stay at regular places for their daily activities. This paper presents a novel framework to construct a predictive model by exploiting the regularity of people movement found in the collected joint Wifi/Bluetooth trace. Our obtained predictive model is able to answer three fundamental questions: (1) where the person will stay at a future time, (2) how long she will stay at the location, and (3) who she will meet at a future time.**

In order to construct the predictive model, we first propose an efficient clustering algorithm to cluster Wifi access points in the Wifi trace into clusters and use these clusters to represent locations. Then, we construct a Naive Bayesian classifier to assign these locations to records in Bluetooth trace. The combined Wifi/Bluetooth trace with locations is used to construct the location predictor, stay duration predictor, and people predictor. Finally, we evaluate three predictors over the real Wifi/Bluetooth traces collected by 50 experiment participants in University of Illinois campus from March to August 2010. The results confirm that our predictors provide highly accurate predictions of location, stay duration, and people.

I. Introduction

The ability to correctly predict the movement of people is crucial to the design of efficient data dissemination protocols and to the network resource planning for Infrastructure-based wireless networks, Mobile Ad hoc Networks (MANET), and Delay Tolerant Networks (DTN). While predicting the movement of a person, we are seeking answers to three fundamental questions: (1) where will the person stay at a future time? (location), (2) How long will she stay at the location? (stay duration), and (3) Who will she meet at a future time? (people/social contact). Providing answers to these questions altogether is challenging due to the dynamic nature of people movement and the lack of a large-scale people movement trace to construct a predictive model, which can predict people movement with a high precision. Nevertheless, there have been several efforts in constructing models to predict people movement. These approaches used past data to construct the model to predict the future movement since it is well known that the daily movement of people exhibits a high degree of repetition [1].

The first class of prediction methods focused on predicting location of people movement [2], [3], [4], [5], [6], [7], [8], which essentially only answer the first question above. In particular, a large number of previous papers used the association trace between the laptop/PDA and the Wifi access points (i.e., WLAN trace) to derive and evaluate their location predictors [6], [7], [9]. However, there was a fundamental weakness of using WLAN trace in constructing location predictor. The reason was that the laptop user did not always turn on the laptop and did not carry it with her all the time. Moreover, a normal laptop user usually turned on her laptop and left it on her office desk when she was doing other things (e.g., had lunch with friends, had meetings with colleagues, or went to exercise at the gym). So, the collected associations of laptops and the Wifi access points could be used to understand the wireless usage rather than to predict the location of people. Other previous projects used cellular data trace to construct the location predictor [4], [3], [5], [10], in which the location was inferred from the cellular ID of the cellular base station. However, since the transmission range of the cellular base station was ranging from several hundred meters (e.g., 500 m) to kilometers (e.g., 30 km), the location predictor derived by this inferred location might not provide needed fine granularity and accuracy.

The second class of prediction methods answer the first two questions above by providing predictions for both location and stay duration at the location [11]. For this paper, Lee and Hou modeled user mobility by a semi-Markov process and devised a timed location prediction algorithm that predicted the future access point (i.e., location in the paper's context) of the user and the association duration. Since the model was constructed and evaluated by WLAN trace, it suffered from the same fundamental weaknesses as discussed in the previous paragraph. However, this paper has been the only method so far which could provide predictions for both location and stay duration.

Previous works only answered the first two questions about location and stay duration of people movement since they lacked of ad hoc contact traces, which could be used to infer social contacts and answer the third question. Recently, there have been several projects collecting ad hoc contact traces using portable experiment devices such as iMote, cellphone, PDA [12], [13], [14], [15], [16], [17]. Due to the limitation of battery and the hardware capability of the experiment devices, only the Bluetooth ad hoc contacts were collected by these projects. Moreover, the scale of these experiments is much smaller in the number of participants and shorter in the experiment duration than those of WLAN experiments in [6], [7]. More importantly, these collected traces did not have the location information and thus could not be used to answer the first two questions above.

In our recent work [18], we designed and implemented a novel scanning system named UIM [1] on Google Android phone to collect both MAC addresses of Wifi access point and Bluetooth ad hoc contacts. The UIM system was then deployed to Google phones carried by experiment participants in University of Illinois campus from March to August 2010. Then, collected MAC addresses of Wifi access point were used to infer location and collected MAC addresses of Bluetooth devices were used to infer social contact[2]. To the best of our knowledge, we are the first to collect both location and social contacts in one data set.

In this paper, we propose a novel framework, which exploits the regularity of people movement found in the joint Wifi/Bluetooth data set to construct a predictive model to predict the location, stay duration, and people for a future time. Particularly, we first propose an efficient clustering algorithm to cluster Wifi access points in the Wifi trace into clusters and use these clusters to represent locations. Then, we present a Naive Bayesian classifier to assign these locations to records in Bluetooth trace. The combined Wifi/Bluetooth trace is used to construct the location predictor, stay duration predictor, and people predictor. Finally, we evaluate the predictors over the real Wifi/Bluetooth traces collected by 50 experiment participants. The results confirm that our framework successfully constructs a predictive model, which could provide highly accurate prediction of location, stay duration, and people altogether for a future time.

This paper is organized as follows. We first present the overview of UIM scan system, its collected joint Wifi/Bluetooth trace, and the overview of the predictive model in Section II. Then, we present a clustering algorithm to cluster Wifi access points in the Wifi trace into locations in Section III. These locations will be assigned to records in Bluetooth trace in Section IV. Then, the Bluetooth and Wifi traces with location will be combined into one combined set, which is used to construct location predictor, duration predictor, and people predictor in Section V. Finally, we evaluate our predictors in Section VI and conclude the paper in Section VII.

II. Overview of UIM Scanning System and Predictive Model

In this section, we present the overview of UIM scanning system, its collected joint Wifi/Bluetooth trace, and the overview of the predictive model.

A. UIM: A Joint Wifi/Bluetooth Scanning System

As presented in our previous work [18], UIM is a novel system running on Google Android phones. To the best of our knowledge, UIM is the first system collecting both MAC addresses of Bluetooth[3]-enabled devices and MAC addresses of Wifi access points in proximity of the experiment phone. UIM has two scanners as presented below.

The *BT scanner* periodically captures the MAC addresses of the Bluetooth-enabled devices in proximity of the experiment phone and the scan time. Let δ_B denote the scan period of the BT scanner and "BT MAC" denote the scanned MAC addresses of the scanned devices. We set $\delta_B = 60(s)$ to conserve the phone battery. The trace collected by BT scanner is called "BT trace" and is denoted by B. Notice that UIM makes the experiment phones discoverable in the BT channel so that an experiment phone can scan other experiment phones in its proximity.

The *Wifi scanner* periodically captures the MAC addresses of the Wifi access points in proximity of the experiment phone and the scan time. Let δ_W denote the scan period of the Wifi scanner and "Wifi MAC" denote the scanned MAC addresses of the Wifi access points. The trace collected by the Wifi scanner is called the "Wifi trace" and is denoted by W. We set the value of $\delta_W = 30(min)$ since (1) in the university campus environment, people do not move too often and usually stay in the offices or buildings for a long time period (e.g., a class session is usually 50 minutes) and (2) performing Wifi scan on the cell phone is energy-consuming. Detailed discussion of UIM system and measurement results can be found in our previous paper [18].

B. UIM Collected Data Set

We had 100 participants[4] carrying the experiment phones from March to August 2010. Essentially, we had done three rounds of experiment: from beginning of March to end of March, from beginning of April to mid of May, and from end of May to mid of August. So, we did not have altogether 100 participants from March to August; however, many participants participated from one month to two months of experiment. Since our participants used the experiment phones as their daily phones and charged the phones for their personal uses, we obtained a rich set of collected data. The participants included CS faculties, CS staff, and CS grads who usually worked inside our department building named Siebel Center. Meanwhile, CS undergrads took classes in different buildings throughout the university campus. Participants from ECE and ABE (e.g., Department of Agricultural and Biological Engineering) stayed in different buildings from Siebel Center.

For an experiment phone[5] p, let D be the entire collected data set, so $D = W \cup B$. W is a set of multiple Wifi records: $W = \{w_1, w_2, w_3, ..., w_{|W|}\}$. Each record $w_i \in W$ is a tuple

[1] UIM stands for University of Illinois Movement.

[2] We use social contact, contact, people interchangeably.

[3] We use BT and Bluetooth interchangeably.

[4] We obtained the IRB permission at University of Illinois to conduct the experiment.

[5] We use "person" and "phone" interchangeably, "stay duration" and "duration" interchangeably.

Scan Time	Set of Wifi MACs
03/08/10 09:15	a_1, a_3
03/08/10 09:50	a_1, a_5
03/08/10 10:15	a_6, a_8
03/08/10 13:50	a_4, a_9
03/14/10 08:15	a_1, a_3

Table I
EXAMPLE OF A WIFI TRACE W

Scan Time	Set of BT MACs
03/08/10 09:15	u_1, u_3
03/08/10 09:16	u_1, u_3
03/08/10 09:17	u_1
03/08/10 13:50	u_4, u_9
03/14/10 08:14	u_1, u_3, u_8

Table II
EXAMPLE OF A BT TRACE B

in the format of $w_i =< A_i, t_i >$, where A_i a set of Wifi MACs returned from one Wifi scan and t_i is the time of that Wifi scan. So, we have $A_i = \{a_1, a_2, ..., a_j, ...\}$, in which a_j is the j^{th} Wifi MAC scanned by the Wifi scanner of p during the entire experiment period. Table I shows an example of Wifi trace W in which each row is one Wifi record w_i. Let W_A be the set of all Wifi MACs scanned by the Wifi scanner for the entire experiment period. For the Table I, $W_A = \{a_1, a_3, a_4, a_5, a_6, a_8, a_9\}$.

Similarly, B is a set of multiple BT records: $B = \{b_1, b_2, b_3, ..., b_{|B|}\}$. Each record $b_i \in B$ is a tuple in the format of $b_i =< U_i, t_i >$, where U_i a set ofBT MACs returned from one BT scan and t_i is the time of that BT scan. So, we have $U_i = \{u_1, u_2, ..., u_j, ...\}$, in which u_j is the j^{th} BT MAC scanned by the BT scanner of p during the entire experiment period. Table II shows an example of BT trace B in which each row is one BT record b_i. Let B_A be the set of all BT MACs scanned by the BT scanner for the entire experiment period. For the Table II, $B_A = \{u_1, u_3, u_4, u_8, u_9\}$. Notice that since the Wifi scanner and BT scanner run concurrently, the scan times of records in W and B overlap.

C. Overview of the Predictive Model

Given the data set D of a person p, we first use MAC addresses of the Wifi access points to represent locations since in reality the wifi MAC access points are usually associated with the physical buildings and can be used as the landmarks to identify buildings/locations [19]. Then, we use BT MACs to infer social contacts to predict people. The combined data set of Wifi and BT trace is used to predict location, stay duration, and people.

Figure 1 shows steps to construct the predictive model from the joint Wifi/Bluetooth trace D. In the first and the second steps, we cluster Wifi records in W into clusters and use these clusters to present locations (see Section III). Then, in step 3 and 4, we construct a Naive Bayesian classifier to

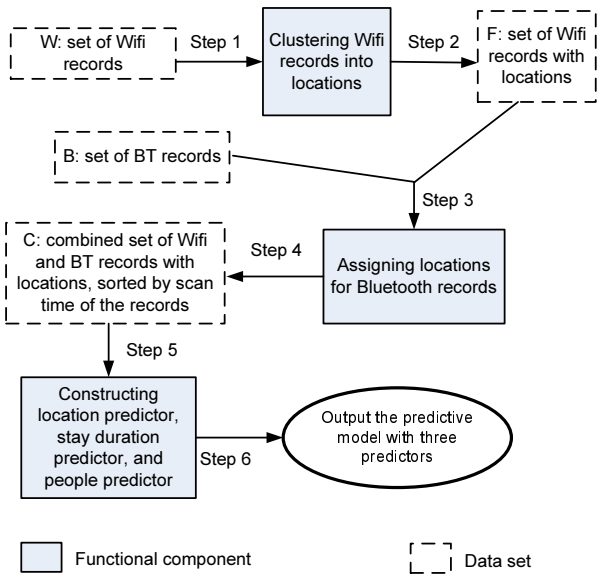


Figure 1. Steps to construct the Predictive Model

assign locations for records in BT trace B and combine the BT trace and Wifi trace (both with assigned locations) into the combined set C (see Section IV). In step 5 and 6, the combined set C is used as the input to construct location predictor, duration predictor, and people predictor (see Section V). Next, we present the construction of our predictive model.

III. CLUSTERING WIFI RECORDS INTO LOCATIONS

This section presents an efficient algorithm to cluster Wifi records of W into clusters and use these clusters to represent locations. We call our clustering algorithm "UIM Clustering" algorithm. This section focuses on step 1 and 2 in Figure 1.

A. UIM Clustering Algorithm Overview

There are several challenges in obtaining location from W. First, the Wifi wireless scanning range of the phone varies from 100 to 200 meters, depending on various factors such as weather, obstacles. So, although the phone stays in one fixed position inside the same building, it may obtain different results for different scans. Previous work [20], [21] used the similarity of signal strength among scanned Wifi MACs to identify locations. However, these approaches suffered from environmental factors since the fluctuation of Wifi signal depends on temperature, obstacles, etc. Second, there are cases when the phone is in the middle of two adjacent buildings, the scanned result might be partially overlapped with the scanned results obtained when the phone stays inside either of the buildings. Fortunately, in reality the movement pattern of people is relatively regular as they tend to stay more frequently at their regular places. So, if two Wifi MACs a_1, a_3 appear together more frequently than two Wifi MACs a_1, a_5 in the Wifi trace W, then it is likely

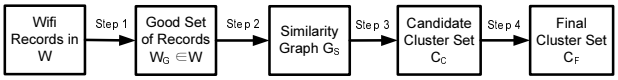


Figure 2. Execution of UIM Clustering algorithm

that a_1, a_3 stay close in a physical building. That means, it is better to group a_1 and a_3 into the same location than a_1 and a_5. This observation motivates us to design a new clustering algorithm to cluster Wifi MACs into locations. For the clustering algorithm, we do not use the scan time and only use the set of Wifi MACs of the Wifi records. We thus call A_i the record i^{th} of the Wifi trace W [6].

For our clustering algorithm, we first define *location as a unique set of Wifi MACs, which appear **frequently** together in the records of* W. In Table I, the pair a_1, a_3 appears twice together while a_1, a_5 appears once. So, we say a_1, a_3 appear together more frequently in W than a_1, a_5.

Figure 2 shows the execution block diagram of the UIM Clustering algorithm. In step 1, given the records in W, we obtain the sub set of *good* records $W_G \subset W$ [7]. In step 2, we measure the similarity between all pairs of records of W_G and construct a similarity graph G_S, in which each vertex of G_S is a record of W_G. In step 3, we apply the Star Clustering algorithm [22] to cluster vertices into a set C_C of candidate clusters. Finally, candidate clusters are merged based on their similarity measures to obtain the set C_F of final clusters. Each cluster in C_F can be used to represent one location. Table III represents major notations used by the UIM Clustering algorithm.

Name	Description
W_G	Set of good records of W, $W_G \subset W$
V_{A_i}	The binary bit vector of A_i, $\|V_{A_i}\| = \|W_A\|$
W'_G	Set of binary vectors: $V_{A_i} \in W'_G$ where $A_i \in W_G$
G_S	The similarity graph: $G_S =< V_S, E_S >$
C_C	Candidate Cluster Set obtained from G_S
$V^S_{C_i}$	Signature vector of cluster $C_i \in C_C$
C_F	Final Cluster Set obtained from C_C
θ	The similarity threshold

Table III
MAJOR NOTATIONS USED BY UIM CLUSTERING ALGORITHM

B. Obtaining the Good Set of Records W_G

This Section focuses on the Step 1 in Figure 2. First, we define *a good record as a record that consists of Wifi MACs appearing **frequently** together in the records of* W. We determine if a record $A_i \in W$ is a good record as follows: for each pair of Wifi MACs $(a_j, a_k) \in A_i$, we calculate the support value $s_{j,k}$, which represents how frequently the pair (a_j, a_k) appears together in the same records of W:

$$s_{j,k} = \frac{c(a_j, a_k)}{min\{c(a_j), c(a_k)\}} \quad (1)$$

In Equation 1, $c(a_j)$ is the number of records $A_i \in W$ in which $a_j \in A_i$, the same applies for $c(a_k)$. $c(a_j, a_k)$ is the number of records $A_i \in W$ in which $a_j \in A_i, a_k \in A_i$. Intuitively, $s_{j,k}$ is similar to the notion of support value in Frequent Item Set Mining literature [23]. For the denominator of Equation 1, we have min of $c(a_j)$ and $c(a_k)$ since we are interested in the Wifi MAC which appears in less number of records and the association of this Wifi MAC with the other one. This min value represents the coexistence of the two Wifi MACs in the records of W. We have $s_{j,k} \in [0, 1]$ and the greater value of $s_{j,k}$ means the two Wifi MACs appear together in the same records of W more frequently.

Let $|A_i|$ be the number of unique Wifi MACs of the record A_i. For each $A_i \in W$, we have $\binom{|A_i|}{2}$ pairs of Wifi MACs, thus we have $\binom{|A_i|}{2}$ support values calculated from the Equation 1. These values constitutes a distribution. Let λ_A and ξ_A be the mean and standard deviation of this distribution. If A_i has only one Wifi MAC, then $\lambda_A = 1, \xi_A = 0$. Intuitively, we prefer a greater value of λ_A since it means A_i contains Wifi access points that often appear together in the records of W. Also, we prefer a smaller value of ξ_A since it means the support values stays in a small range. So, for each record A_i, we calculate the ratio $\frac{\xi_A}{\lambda_A}$ to: (1) select good record whose Wifi MACs appear together frequently in the same records of W, and (2) remove the bad records consisting of wifi MACs, which do not frequently appear together in records of W. Let F_W be the set of ratios $\frac{\xi_A}{\lambda_A}$ of all records of W. We then sort F_W increasingly and create the set W_G of good records from W using F_W as presented in the Algorithm 1.

Algorithm 1 Obtain W_G from W using F_W, W_A

```
Input: W, F_W, W_A
Output: W_G
BEGIN
W_C = ∅; # set of wifi MACs belonging to records in W_G
for each ratio ξ_A/λ_A ∈ F_W do
    Find the corresponding record A_i ∈ W;
    M_A = set of Wifi MACs of A_i;
    if |W_C ∪ M_A| > |W_C| then
        W_G = W_G ∪ A_i;
        if |W_C| == |W_A| then
            return W_G;
        end if
    end if
end for
END
```

The intuition of the Algorithm 1 is as follows. We always prefer records with smaller ratio $\frac{\xi_A}{\lambda_A}$. Since we need to

[6] In other sections, we use w_i to represent the record i^{th} of W.

[7] see Section III-B for definition of good record

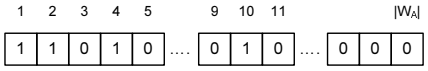


Figure 3. Bit vector V_{A_i}, with $A_i = \{a_1, a_2, a_4, a_{10}\}$

consider all Wifi MACs in W, one record is only useful if adding its Wifi MACs to W_C increases the size of W_C; otherwise, the record is filtered out. Doing this, we reduce the number of records and remove most of the noisy data resulting from records whose wifi MACs do not appear together frequently in records of W. As a result, the set W_G is good for the clustering algorithm in the next step.

C. Constructing Similarity Graph G_S

This section focuses on the Step 2 in Figure 2. Given the good set W_G, we map each record $A_i \in W_G$ into a binary bit vector V_{A_i} as follows. If the Wifi MAC $a_j \in A_i$, then the j^{th} bit of the vector V_{A_i} is set to 1, $V_{A_i}[j] = 1$; otherwise, $V_{A_i}[j] = 0$. Figure 3 shows an example of the binary bit vector.

Let W'_G be the set of binary vectors obtained from all records $A_i \in W_G$. Then, we use the Tanimoto coefficient [24] (a special form of cosine similarity) to calculate the similarity between a pair of vectors $V_p \in W'_G, V_q \in W'_G$:

$$T_{p,q} = \frac{V_p \cdot V_q}{||V_p||^2 + ||V_q||^2 - V_p \cdot V_q} \quad (2)$$

In Equation 2, $T_{p,q}$ is the similarity measure of V_p and V_q. Next, we construct the similarity graph $G_S =< V_S, E_S >$, in which each vector $V_p \in W'_G$ is considered a vertex $v_p \in V_S$, so we have: $|V_S| = |W'_G|$. For a pair of vertices $v_p, v_q \in V_S$, the edge (v_p, v_q) exists (i.e., $(v_p, v_q) \in E_S$) if $T_{p,q} \geq \theta$. θ determines the topology of G_S and has important impacts on the clustering results (see Section III-F).

D. Obtaining Candidate Cluster Set C_C

This section focuses on Step 3 in Figure 2. Particularly, we apply the Star Clustering algorithm [22] to cluster vertices of G_S into clusters. We opt for Star Cluster algorithm since it does not require a pre-defined number of clusters like other clustering algorithms such as partition clustering (e.g., k-means) or hierarchical clustering (e.g., DIANA). Start Clustering thus fits very well to our context since we do not know in advance the number of locations we can obtain from the set of records in W. The Star Clustering algorithm works as follows. We first sort the vertices decreasingly according to their node degrees. Then, we scan the sorted list of vertices, for each vertex v_p if v_p is not in any clusters, v_p is considered a center of a new cluster. For each neighboring vertex v_q of v_p, if v_q does not belong to any clusters, v_q is included in the cluster centered at v_p. The process continues until all the vertices belong to clusters. We denote this set of clusters *the candidate cluster set* C_C.

E. Obtaining Final Cluster Set C_F

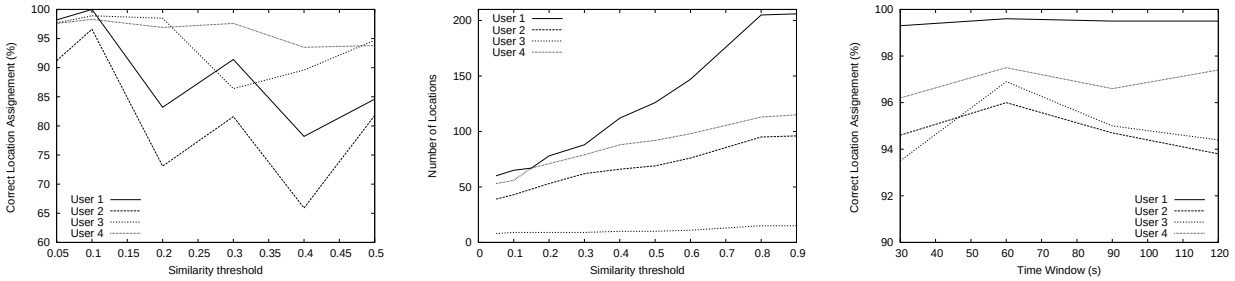
This section focuses on the Step 4 in Figure 2. For a cluster $C_i \in C_C$, C_i consists of a set of vertices, each vertex is a binary vector representing a record $w \in W_G$. Let $V^S_{C_i}$ be the signature vector of the cluster C_i. $V^S_{C_i}$ is obtained by applying the OR bitwise operation over all the binary vectors of C_i. Intuitively, the signature vector $V^S_{C_i}$ represents the set of Wifi MACs, which belong to the cluster C_i. Thus, the signature vector $V^S_{C_i}$ can be used to uniquely distinguish clusters in C_C. Then, we use the signature vectors to merge cluster $C_1 \in C_C$ into cluster $C_2 \in C_C$ if C_1 is a sub cluster of C_2. Formally, C_1 is merged into C_2 if $V^S_{C_2} = (V^S_{C_1}\ OR\ V^S_{C_2})$. So, we have the final set of clusters C_F, in which each cluster $C_j \in C_F$ can be used to represent one particular location.

Given the final cluster set C_F, we classify all Wifi records $A_i \in W$ into clusters in C_F as follows. Each record $A_i \in W$ is classified to the best matched cluster $C_i \in C_F$ based on the similarity measure between V_{A_i} and $V^S_{C_i}$ calculated by Equation 2. The output of this step is the set F of all Wifi records with assigned locations as shown in Table IV.

F. Setting value of Similarity Threshold θ

The value of θ decides the topology of G_S and thus has crucial impacts on clustering results. To obtain the value of θ for our clustering algorithm, we first select 4 different participants and create for each of them a development set W_D, which consists of 64 Wifi records scanned in two different days. Then, we ask the participants to manually label the location for their Wifi records (e.g., Long's home, Quang's home, Klara's office, etc.). For each value of $\theta \in [0.05, 0.9]$, we perform following steps. For each pair of records $(A_1, A_2) \in W_D$ (i.e., W_D has $\binom{64}{2}$ pairs), we check cluster ids of A_1 and A_2 in F and compare these cluster ids with the labeled locations in W_D. A location assignment made by UIM Clustering algorithm is correct if: (1) A_1 and A_2 have the same labeled location in W_D and they are assigned into the same cluster in F, or (2) A_1 and A_2 have different labeled locations in W_D and they are assigned into different clusters in F. Figure 4(a) shows the percentage of correct classification the clustering algorithm makes when θ varies from 0.05 to 0.9. The best value of all people is 0.1, in which the correct prediction for all 4 people is greater than 96%. When $\theta = 0.05$, clusters are merged into big cluster; or nearby locations are merged into one location, it may incur "too big locations" and result in incorrect location assignment. In contrast, when θ increases, nearby clusters are separated. Thus, for a high value of θ (e.g., $\theta > 0.1$), two records of the same location may be assigned into different clusters.

To understand the sensitivity of θ, we varies θ in the range of [0.05, 0.9] and count the number of unique locations each of the four above people visited during their entire



(a) $\theta = 0.1$ assigns most correct locations (b) # of locations increases when θ increases (c) $\alpha = 60(s)$ assigns most correct locations

Figure 4. Sensitivities of θ and α

Scan Time	Set of Wifi MACs	Location
03/08/10 09:15	a_1, a_3	L_1
03/08/10 09:50	a_1	L_1
03/08/10 10:15	a_6, a_8	L_5
03/08/10 13:50	a_4, a_9	L_8
03/14/10 08:15	a_1, a_3	L_1

Table IV
EXAMPLE OF THE SET F

experiment periods. Figure 4(b) shows that the number of clusters increases nearly linearly when θ increases from 0.05 to 0.9. This result is expected since for greater value of θ, G_S is sparser, so the cluster size is smaller and the number of cluster is bigger. So, we use $\theta = 0.1$ to cluster Wifi records into location in UIM Clustering algorithm and to evaluate the predictive model in Section VI.

Name	Description
μ	# of unique BT MACs collected by all phones
F	Set of Wifi records with assigned locations
M	Set of BT records created by F, B and α
C	Combined set of Wifi and BT with locations
α	The time window in second
β_{min}	The threshold to assign "Unknown" location
ν	The type of day, including Weekend and Weekday
σ	Time slot size of a day, $\sigma \in [1, 2, 3, ..., 24]$

Table V
MAJOR NOTATIONS OF THE PREDICTIVE MODEL

IV. ASSIGNING LOCATIONS FOR BLUETOOTH RECORDS

Although all records in F are assigned locations, they do not provide needed granularity of people movement since the Wifi scanner scans each every 30 minutes. During this period, a person may move to different locations. Meanwhile, our BT scanner scans every minute and provides a rich set of BT MACs. Our goal is to assign locations to BT records and thus obtain the needed granularity of people movement for our predictive model. The first step toward this goal is to map Wifi records and BT records using a time window α. This section focuses on step 3 and 4 in Figure 1. Table V presents the major notations used in following sections.

A. Mapping between Wifi Records and BT Records Using Time Window α

For a record $w \in F$ whose scan time is t and location is L, we know that the phone p is at the location L at time t. From our data set, we observe that during the time window $[t - \alpha, t + \alpha]$, if α is short enough, the person usually stays at L. Therefore, we can assign the location L of w to all BT records $b \in B$ whose scan time is within the time window $[t - \alpha, t + \alpha]$. Let M be the set of all BT records $b \in B$, which are assigned locations using the time window α and scan times of all Wifi records in F. Table VI shows an example of the set M, which is created through the maping between B and F using the time window α. α plays an important role in our predictive model thus we will present a separate section (i.e., Section IV-C) on how to set the value of α in order to construct our predictive model.

Scan Time	Set of BT MACs	Location
03/08/10 09:15	u_1, u_3	L_1
03/08/10 09:16	u_1, u_3	L_1
03/08/10 13:50	u_4, u_9	L_8
03/14/10 08:14	u_1, u_3, u_8	L_1

Table VI
THE SET M CREATED THROUGH THE MAPPING BETWEEN SET F AND SET B WITH $\alpha = 60(s)$

B. Assigning Locations for Bluetooth Records

In this section, we construct a Naive Bayesian classifier N_B to predict the locations of all BT records in B. Basically, we use the set M to train the Naive Bayesian classifier N_B and then use N_B to assign locations to all records $b \in B$.

1) Training the Naive Bayesian Classifier N_B: For a BT record $b_i \in B$, the probability that b_i belongs to a location L_k is calculated by Bayesian Theorem as follows:

$$P(L_k|b_i) = \frac{P(b_i|L_k)P(L_k)}{P(b_i)} \quad (3)$$

Then, b_i belongs to the location L_{b_i}, which is calculated as follows:

$$L_{b_i} = \arg\max_i \frac{P(b_i|L_k)P(L_k)}{P(b_i)} \quad (4)$$

Since $P(b_i)$ is the same for all locations L_k, we need to calculate $f(L_k) = P(b_i|L_k)P(L_k)$ to find L_{b_i}. To calculate $P(b_i|L_k)$, we assume that for $u_1 \in b_i$ and $u_2 \in b_i$, u_1 and u_2 are conditionally independent. In other words, we assume that u_1 and u_2 appear conditionally independent in the proximity of the experiment phone when they are scanned (and b_i is created) by the BT scanner. This assumption usually holds in reality since people (with their Bluetooth-enable devices) appear at locations independently. With this assumption, $f(L_k) = \Pi_{u_j \in b_i} P(u_j|L_k)P(L_k)$ and the Equation 4 becomes:

$$L_{b_i} = \arg\max_i \ f(L_k) \tag{5}$$

The set M is used to calculate $f(L_k)$ in Equation 5 as follows. $P(L_k) = \frac{c(L_k)}{|M|}$, where $|M|$ is the size of M, and $c(L_k)$ is the number of records $b_i{}' \in M$, in which the location of $b_i{}'$ is L_k. For $P(u_j|L_k)$, we have:

$$P(u_j|L_k) = \frac{c(u_j)}{c(L_k)} \tag{6}$$

In Equation 6, $c(u_j)$ is the number of records $b_i{}' \in M$, in which the location of $b_i{}'$ is L_k and $u_j \in b_i{}'$. Apply Equation 5 and Equation 6 for all records of M, we have the trained classifier N_B.

2) Applying Additive Smoothing Technique: In section IV-A, since we only use a small time window α to create M, M does not cover all BT MACs in B. Thus, applying the trained classifier N_B from Section IV-B1 for a record $b_i \in B \setminus M$, the value $c(u_j)$ in Equation 6 might be 0 and cancel out the value of $P(u_k|L_k)$ of BT MACs $u_k \in b_i$ (i.e., $j \neq k$) in Equation 5. To avoid this, we apply the Additive Smoothing technique [25] for the Equation 6 as follows:

$$P(u_j|L_k) = \frac{c(u_j) + 1}{c(L_k) + \mu} \tag{7}$$

In Equation 7, μ is the number of unique BT MACs collected by all participants for the entire experiment period. Adding μ to the denominator of Equation 7 means we take into account all possible BT MACs in calculating the probability of the BT MAC u_j. With Equation 7, $P(u_j|L_k) \neq 0$ for all u_j and we have:

$$f(L_k) = \Pi_{u_j \in b_i} \frac{c(u_j) + 1}{c(L_k) + \mu} P(L_k) \tag{8}$$

So, we have a new trained classifier N'_B by applying Equation 5 and Equation 8 for all records of M.

3) The "Unknown" Location: Applying N'_B to assign locations to BT records $b_i \in B$, we encounter records b_i whose value of $f(L_{b_i})$ (i.e., calculated by Equation 4) is extremely small. These records b_i are scanned in the middle of two consecutive Wifi scans (i.e., $\delta_W = 30$ minutes) when the phone carrier moves to another location, which is not captured by the Wifi scanner. Therefore, assigning any known location to b_i results in a wrong assignment. To avoid this, we define a new location named "Unknown" location and assign the "Unknown" location to b_i.

The next question is "How small the value of $f(L_{b_i})$ is so that the record b_i is assigned to "Unknown" location. To answer this question, we use Equation 8 to calculate $f(L_{b_i{}'})$ for all records $b_i{}' \in M$. Let $\beta_{min} = min_{b_i{}' \in M} f(L_{b_i})$. We then use β_{min} as the threshold value to assign "Unknown" location to a BT record $b_i \in B$. The intuition is as follows. We assume that records in M are "good records" whose locations are assigned correctly. So, the minimum value of $f(L_{b_i{}'})$ of all records $b_i{}' \in M$ represents the cutoff value for all records whose locations are assigned correctly. For a record $b_i \in B$, we have:

$$L_{b_i} = \begin{cases} \arg\max_i \ f(L_k) & \text{if } f(L_{b_i}) \geq \beta_{min} \\ \text{"}Unknown\text{"} & \text{otherwise} \end{cases} \tag{9}$$

Equation 9 means b_i will be assigned L_{b_i} location only if $f(L_{b_i}) \geq \beta_{min}$. Otherwise, b_i will be assigned the "Unknown" location. Although this approach seems to be conservative in assigning correct locations to BT records, it does provide good result in our evaluation of the predictive model in Section VI. Let B' be the set of all records in B, which are assigned locations. Then, we have $C = B' \cup F$, a combined set of Wifi and BT records, in which all records are assigned locations. Then, we sort C increasingly according to the scan time of its records and use C as the input to construct our predictors in Section V.

C. Setting Value of Time Window α

As we presented in Section IV-A, the value of α decides the mapping between Wifi records and BT records and the size of set M, which is used to train the Naive Bayesian classifier N_B. Therefore, it is important to have the right value of α. In this section, we use the same technique in Section III-F to set value for α.

Particularly, we select 4 participants and for each of them, we create a set B_D of BT records of two days and ask the participants to manually label locations for records in his B_D. For the records that the participants do not know the location, they can mark it "Unknown" location. For each pair of records $(b_1, b_2) \in B_D$ (i.e., one B_D has $\binom{960}{2}$ pairs), we check locations of b_1 and b_2 in C and compare these locations with the labeled locations in B_D. Figure 4(c) shows that when $\alpha = 60(s)$, the set C outputted by Naive Bayesian classifier obtains the best location assignment, in which the correct prediction for all 4 people is greater than 95%. With $\alpha = 30(s)$, the set M consists of too few records to train a good Naive Bayesian classifier. Meanwhile, $\alpha > 60(s)$ is too large a time window, which maps many BT records from BT trace into one Wifi record in Wifi trace and incur noisy data in the set M since records may be assigned to wrong locations if they fall into this big time window. The trained

classifier N_B then performs worse than that with $\alpha = 60(s)$. So, we use $\alpha = 60(s)$ to create the mapping M between Wifi trace and BT trace, and to evaluate the performance of our predictive model in Section VI.

V. Constructing Location Predictor, Duration Predictor, and People Predictor

Given the combined set C of Wifi and BT records, we construct the location predictor, duration predictor, and people predictor. Basically, this section focuses on step 5 and step 6 in Figure 1.

To construct our predictors, we use two parameters: type of day and time slot. Let ν be the "type of day" and σ be the "time slot". Particularly, we classify days into two types: weekend and weekday, so $\nu \in \{Weekday, Weekend\}$, and divide time of a day into time slot of 1, 2, 4, etc. hours, so $\sigma = \{1, 2, 3, ..., 24\}$. The motivation for the use of these parameters is that people may have different movement behaviors and meet different people for the weekday and weekend. For each record $r \in C$, we map r's scan time into type of day ν and time slot σ. Table VII shows an example of a combined set C in which its records are mapped into type of day and time slot of size $\sigma = 2$ hours. The combined set C in this new format is used to construct our predictors.

ν	σ	**Scan Time**	**Loc**	**Set of BT MACs**
Weekday	08-10	03/08/10 09:15	L_1	u_1, u_3
Weekday	08-10	03/08/10 09:16	L_1	u_1, u_3
Weekday	08-10	03/08/10 09:17	L_1	u_1, u_8
Weekday	12-14	03/08/10 13:50	L_8	u_4, u_9
Weekend	08-10	03/14/10 08:12	L_8	u_4, u_{12}
Weekday	08-10	03/15/10 09:47	L_1	u_1
Weekend	14-16	03/20/10 15:23	L_3	u_{15}

Table VII
Example of Combined Set C

For our predictors, the input query is a record X in the format of $X = \{\nu_1, \sigma_1\}$, in which ν_1 represents the type of day and σ_1 represents the time slot. The output will be location the person stays at, duration the person stays at the location, and people the person meets for the type of day ν_1 and during time slot σ_1.

A. Location Predictor

We use the Naive Bayesian classifier to predict the location of the person as follows.

$$L_X = \arg\max_i \{P(\nu = \nu_1|L_k)P(\sigma = \sigma_1|L_k)P(L_k)\} \quad (10)$$

The Equation 10 outputs the most likely location L_X for the input query X. Moreover, the Equation 10 can be easily customized to return the top-k of the most likely locations for the input query X. In this case, L_X is the set of top-k most likely locations and we have a *top-k location predictor*.

B. Duration Predictor

The duration predictor is constructed based on the location predictor. In other words, if the location predictor returns a top-k locations, the duration predictor will return the predicted stay duration for each of k locations.

We first define the *"stay session at the location L_k" is the continuous time period that the person stays at L_k*. In our context, since the BT scan obtains BT records every minute, the *"stay session at the location L_k in minute" is the size of the set Φ of consecutive records in set C such that for two consecutive records $r_1, r_2 \in \Phi$, the difference of scan time between r_1 and r_2 is exactly 1 minute.* Let $|\Phi|$ denote the session length of one stay session of L_k.

The next step is to obtain all session lengths $|\Phi|$ of each location L_k returned by the top-k location predictor for the input query $X = (\nu_1, \sigma_1)$. So, for each location L_k, we create a subset $C' \subset C$ consisting of records whose type of day, time slot, and location are ν_1, σ_1, and L_k, respectively. Then, we calculate the session lengths for L_k using the above definition. Let Γ_i be the set of all stay session lengths for the location L_k obtained from set C', $\Gamma_i = \{\Phi_1, \Phi_2, \Phi_3, ..., \Phi_{|\Gamma_i|}\}$, Γ_i forms a distribution of session lengths. Let λ_i and ξ_i denote the mean and standard deviation of this distribution. We then calculate λ_i and ξ_i and output them together with the location L_k in Equation 10 as the output of the duration predictor. For example, the location L_1 in Table VII has $\Gamma_1 = \{3, 1\}$, here $|\Phi_1| = 3$ and $|\Phi_2| = 1$ (Φ_1 consists of the first three records).

C. People Predictor

Notice that in order to construct the people predictor, we assume that each BT MAC scanned by the BT scanner is associated with a distinct person. We apply the Naive Bayesian classifier to find the most likely people the person will meet:

$$U_X = \arg\max_j \{P(\nu = \nu_1|u_j)P(\sigma = \sigma_1|u_j)P(u_j)\} \quad (11)$$

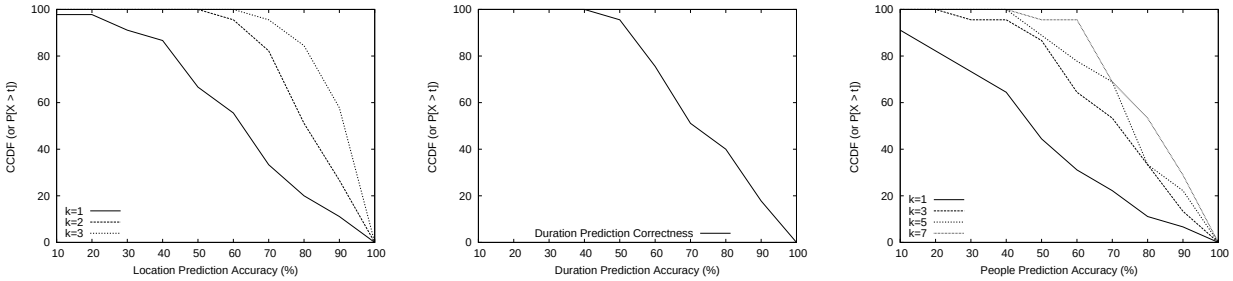
The Equation 11 outputs the most likely person U_X for the input query X. Moreover, the Equation 11 can be easily customized to return the top-k of the most likely people for the input query X. In this case, U_X is the set of top-k most likely people and we have a *top-k people predictor*.

VI. Evaluation of the Predictive Model

In this section, we evaluate the correctness and performance of location predictor, duration predictor, and people predictor.

A. Settings

From March to August 2010, we had 100 participants participating in our experiment. We examine the traces and select a set of 50 good traces collected by 50 participants.



(a) Location Prediction (b) Duration Prediction (c) People Prediction

Figure 5. Correctness of Prediction Model

Each of these traces is from 28 to 55 days long. Let Δ_i be the Wifi/Bluetooth trace of the i^{th} participant in 50 participants: $\Delta_i = W_i \cup B_i$, where W_i is the Wifi trace and B_i is the BT trace. For i^{th} participant, we first apply the UIM Clustering Algorithm over W_i to obtain locations. Then, we apply steps in Section IV to assign locations to records in B_i. For each user, we divide the set C_i into two distinct subsets called training set Ψ_i and testing set Ω_i, in which $\Psi_i \cap \Omega_i = \emptyset$. The training set Ψ_i has 80% of records in C_i and Ω_i has 200 records randomly picked from the set $C_i \setminus \Psi_i$. We use Ψ_i to training the three predictors (location, stay duration, and people) in Section V and use Ω_i to test these predictors.

The reason we use part of C_i (i.e., $\Omega_i \subset C_i$) as the testing set is as follows. First, we use development sets manually labeled by 4 experiment participants to set values of θ and α. With $\theta = 0.1$, the UIM Clustering algorithm obtains more than 95% correct location assignments for these 4 people as shown in Figure 4(b). Similarly, with $\alpha = 60(s)$, the Naive Bayesian classifier obtains more than 96% correct location assignments for these 4 people as shown in Figure 4(c). Therefore, we believe that setting $\theta = 0.1$ and $\alpha = 60(s)$ is suitable for our data set and that the combined set C_i contains records with correct location assignments. As a result, $\Omega_i \subset C_i$ can be used to test the predictors. By default, we set $\theta = 0.1$, $\alpha = 60(s)$, and $\sigma = 2$ hours in the following plots. We run the experiment 10 times[8] and plot the average for each experiment participant.

The input query for our predictors is in the format of $X = \{\nu_1, \sigma_1\}$. For a specific record $r \in C_i$, since r has its scan time, the scan time can be converted to the format of X, which is used as the input for the predictors below.

B. Correctness of the Predictors

1) Location predictor: Let ${L^i}_p$ be the location predictor of the i^{th} experiment participant. For each record $r \in \Omega_i$, we use ${L^i}_p$ to predict the location of r using technique in Section V-A. Let L_r be the location of $r \in \Omega_i$. Since the predictor ${L^i}_p$ can output the top-k most likely locations, let L_{pred} be the set of predicted locations outputted by ${L^i}_p$ so $|L_{pred}| = k$. ${L^i}_p$ makes a correct prediction if $L_r \subseteq L_{pred}$.

Figure 5(a) shows the correctness of ${L^i}_p$ for 50 users with k from 1 to 3. When k increases, the set L_{pred} has more elements, thus the prediction is more likely to be correct, which is confirmed in this figure. Particularly, when $k = 2$, about 80% of nodes have more than 70% correct predictions. When $k = 3$, about 90% of nodes have more than 80% correct predictions. This confirms that the location predictor provides a good performance in predicting the location of a person.

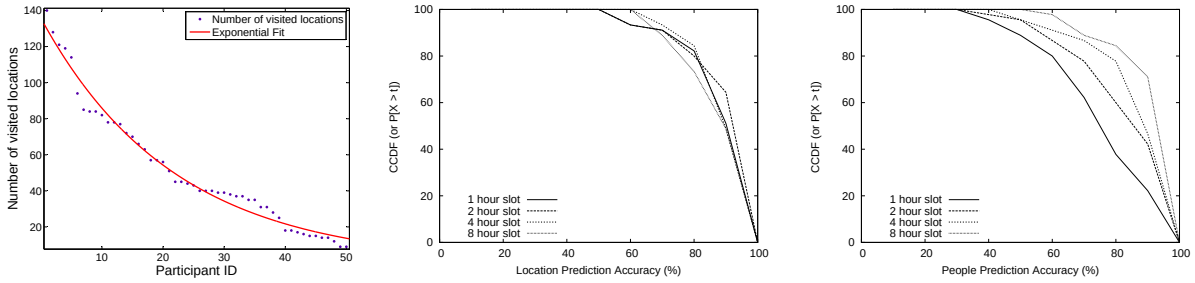
2) Duration predictor: Let ${\Lambda^i}_p$ be the duration predictor of the i^{th} experiment participant. For each record $r \in \Omega_i$, we first convert r into $X = \{\nu_1, \sigma_1\}$. Let λ_{pred} and ξ_{pred} be the mean and standard deviation values return by ${\Lambda^i}_p$ for the input query $X = \{\nu_1, \sigma_1\}$. Then, we use the definition in Section V-B to find the stay session that contains $r \in C_i$. Notice that r should belong to an unique session since r has its own scan time and location. Let Λ_r be the length of the stay duration session that contains $r \in C_i$.

Predicting the stay duration of a person at a location in the future is difficult since the value of stay duration may vary in a wide range. So, we evaluate the correctness of ${\Lambda^i}_p$ as follows: if $\Lambda_r \geq \lambda_{pred} - \xi_{pred}$ and $\Lambda_r \leq \lambda_{pred} + \xi_{pred}$, then ${\Lambda^i}_p$ makes a correct prediction. For this experiment, we use the top-1 location predictor.

Figure 5(b) shows that the duration predictor performs considerably well. Particularly, 80% of nodes obtain about 60% correct prediction and 40% of nodes have about 80% correct prediction. Since the stay duration of people at one location is difficult to predict, we believe this result confirms that the duration predictor can provide a good duration prediction.

3) People predictor: Let ${P^i}_p$ be the people predictor of the i^{th} experiment participant. For each record $r \in \Omega_i$, let P_{pred} be the set of top-k people returned by ${P^i}_p$, so $|P_{pred}| = k$. Since the movement of people is dynamic, it is difficult to predict when people have contacts. Thus, we make ${P^i}_p$ more robust as follows. First, let P_r be the set of people appearing in C_i in the same day and during the

[8]For each time, a new testing set Ω_i is picked randomly.



(a) Number of Locations (b) Impact of Time slot on location (c) Impact of Time slot on people

Figure 6. Predictive Model Evaluation

same time slot σ_1 of r. Second, the predictor ${P^i}_p$ makes a correct prediction if $P_{pred} \cap P_r \neq \emptyset$. The intuition is that ${P^i}_p$ predicts that in the same day of r and during the same time slot of r, P_{pred} is the set of people, in which the person will meet at least one.

Figure 5(c) shows that ${P^i}_p$ performs better when k increases from 1 to 7. With k=7, about 80% of participants can obtain more than 70% correct prediction and about 60% of participants obtains more than 80% correct prediction.

C. Other Characteristics of the Predictive Model

This section presents results to highlight different characteristics of our data set and the robustness of our predictors.

1) Number of locations for all people: In order to known how many locations a participant may visit during the experiment period, for each participant i^{th}, we take the set C_i and count the number of unique locations. Then, we sort the list of participants decreasingly according to their number of visited locations. Figure 6(a) shows that the number of unique locations the participants visited during the experiment period can be fitted by an exponential function $y = ae^{bx}$ in Matlab, where $a = 135.2, b = -0.05023$. This result is important since it gives a concrete model for the number of locations for mobile nodes, which can be used for simulation purpose in mobile networking research. Particularly, instead of taking a random number as the number of locations for a mobile node in simulations, researchers might take a number following an exponential function as shown in Figure 6(a).

2) Impact of σ time slot size: We vary the value of time slot σ and evaluate the performance of our predictors.

Location predictor: Figure 6(b) shows that when the time slot σ varies the performance of location predictor changes slightly. This confirms that the location predictor is robust to the variation of time slot σ. The reason the location predictor stays robust to the variation of σ is that people in university campus do not move very frequently and people usually stay at one location for a long period. Thus, the impact of σ becomes less significant. Notice that for this figure, we use the top-3 location predictor.

People predictor: Figure 6(c) shows that when the time slot σ increases in size, the people predictor performs better. For this figure, we use top-3 people predictor. Particularly, when $\sigma = 8$ hours, about 80% of nodes have 80% correct people prediction. Meanwhile, with $\sigma = 1$ hour, only 40% of nodes have 80% correct people prediction. This is expected since for a bigger time slot σ, we have a bigger subset P_{pred} and a bigger set P_r (as discussed in Section VI-B3), then the prediction is more likely to be correct.

VII. Conclusion

This paper presents a novel framework to construct a predictive model of people movement from the large-scale joint Wifi/Bluetooth trace. Particularly, we first design an efficient clustering algorithm to cluster Wifi access points into locations and apply the Naive Bayesian classifier to assign locations for records in BT trace. Then, the combined trace of Wifi/Bluetooth is used to construct location predictor, duration predictor, and people predictor. Finally, we evaluate our predictors using real movement trace collected by 50 participants in University of Illinois campus from March to August 2010. The evaluation shows that our predictors provide a highly accurate predictions for location, stay duration, and people.

To the best of our knowledge, we are the first to provide a comprehensive predictive model, which can predict location, duration, and people altogether. Since the future knowledge of people movement is fundamental for research in various domains such as wireless networks, HCI, social sciences (e.g., social interaction, social network), etc. , we believe our predictive model is widely applicable. It is well known that the daily movement of people exhibits a high degree of repetition, we believe our framework can be used to derive the predictive model for the joint Wifi/Bluetooth trace collected by different classes of people.

References

[1] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility pattern," *Nature*, vol. 453, pp. 779–782, June 2008.

[2] G. Liu and G. Maguire, "A class of mobile motion prediction algorithms for wireless mobile computing and communications," *Mobile Networks and Applications*, vol. 1, pp. 113–121, June 1996.

[3] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, M. Kyriakakos, and A. Kalousis, "Predicting the location of mobile users: a machine learning approach," in *Proceedings of the 2009 international conference on Pervasive services*, 2009.

[4] K. Laasonen, "Clustering and prediction of mobile user routes from cellular data," in *Proceedings of PKDD*, 2005, pp. 569–576.

[5] P. N. Pathirana, A. V. Savkin, and S. Jha, "Mobility modelling and trajectory prediction for cellular networks with mobile base stations," in *Proceedings of MobiHoc*, 2003.

[6] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive wi-fi mobility data," in *Proceedings of Infocom*, 2004.

[7] R. Jain, A. Shivaprasad, D. Lelescu, and X. He, "Towards a model of user mobility and registration patterns," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, pp. 59–62, 2004.

[8] M. Sun and D. Blough, "Mobility prediction using future knowledge," in *Proceedings of MSWiM*, 2007.

[9] W. Gao and G. Gao, "Fine-grained mobility characterization: Steady and transient state behaviors," in *Proceedings of Mobihoc*, 2010.

[10] M. A. Bayira and M. Demirbasa, "Mobility profiler: A framework for discovering mobility profiles of cell phone users," *Pervasive and Mobile Computing*, 2010.

[11] J.-K. Lee and J. C. Hou, "Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application," in *Proceedings of Mobihoc*, 2006.

[12] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *Proceedings of IEEE Infocom*, 2010.

[13] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proceedings of Infocom*, 2006.

[14] P. Hui, A. Chaintreaum, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the ACM SIGCOMM workshop on Delay-tolerant networking*, 2005.

[15] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," in *Proceedings of CHANTS*, 2006.

[16] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, "User mobility for opportunistic ad-hoc networking," in *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, 2004.

[17] E. P. S. Gaito and G. P. Rossi, "Opportunistic forwarding in workplaces," in *Proceedings of ACM WOSN*, 2009.

[18] L. Vu, K. Nahrstedt, S. Retika, and I. Gupta, "Joint bluetooth/wifi scanning framework for characterizing and leveraging people movement in university campus," in *Proceedings of MSWiM*, 2010.

[19] "Skyhook. http://www.skyhookwireless.com."

[20] J. Krumm and K. Hinckley, "The nearme wireless proximity server," in *In Proceedings of Ubicomp*, 2004.

[21] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," in *In Proceedings of Ubicomp*, 2005.

[22] J. Aslam, K. Pelekhov, and D. Rus, "Static and dynamic information organization with star clusters," in *Proceedings of the Conference on Information Knowledge Management*, 1998, pp. 208–217.

[23] *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann Publishers, 2001, pp. 225–276.

[24] "Tanimoto coefficient, http://en.wikipedia.org/wiki/cosine_similarity."

[25] "Additive smoothing, http://en.wikipedia.org/wiki/additive_smoothing."