3-1-2011

Identifying Unusual Days

Minkyong Kim
IBM Watson Research

David Kotz
Dartmouth College, David.F.Kotz@Dartmouth.EDU

Journal of Computing Science and Engineering
Vol. 5, No. 1, March 2011. pp. 71-84

DOI: 10.5626/JCSE.2011.5.1.071
pISSN: 1976-4677 eISSN: 2093-8020

Identifying Unusual Days

Minkyong Kim
IBM Watson Research, Hawthorne, NY, USA, minkyong@us.ibm.com

David Kotz
Dartmouth College, Hanover, NH, USA, kotz@cs.dartmouth.edu

Received 15 November 2010; Revised 27 February 2011; Accepted 3 March 2011

Pervasive applications such as digital memories or patient monitors collect a vast amount of data. One key challenge in these systems is how to extract interesting or unusual information. Because users cannot anticipate their future interests in the data when the data is stored, it is hard to provide appropriate indexes. As location-tracking technologies, such as global positioning system, have become ubiquitous, digital cameras or other pervasive systems record location information along with the data. In this paper, we present an automatic approach to identify unusual data using location information. Given the location information, our system identifies unusual days, that is, days with unusual mobility patterns. We evaluated our detection system using a real wireless trace, collected at wireless access points, and demonstrated its capabilities. Using our system, we were able to identify days when mobility patterns changed and differentiate days when a user followed a regular pattern from the rest. We also discovered general mobility characteristics. For example, most users had one or more repeating mobility patterns, and repeating mobility patterns did not depend on certain days of the week, except that weekends were different from weekdays.

Keywords: Mobility Characteristics, Wireless Network Trace Study, User Classification

1. INTRODUCTION

As the cost of sensors and storage devices goes down rapidly, ubiquitous applications tend to capture and record a vast amount of data. As one example, digital memories that record everyday life have been an active area of research [Czerwinski et al. 2006; Lamming and Newman 1992]. Some healthcare applications monitor patients around the clock [Wilson and Atkeson 2004]. One key challenge in these systems is how to extract interesting or unusual information. When data is stored, it is hard to anticipate how the data will be searched in the future. Thus, providing appropriate indexes for data at the time when it is stored may not be feasible. Requiring users to annotate the data is inconvenient or even infeasible in certain cases.

In this paper, we propose an automatic way to detect unusual days (i.e., days with unusual mobility patterns) for individual users. In our detection system, we used the locations that a user visited during each day to build mobility profiles dynamically and to classify user days. For the evaluation, we used the location data of wireless

users collected by wireless access points. Although we used wireless location data, which can be collected only where wireless connectivity was available, we expect that many pervasive devices (i.e., smart phones) are or will be equipped with a location-tracking mechanism such as global positioning system (GPS). Our system allows users to easily identify unusual days of their life, detect changes in their mobility patterns, and understand their mobility characteristics. These capabilities will benefit numerous pervasive systems that deal with a large amount of data.

Bush [1945] mentioned a forehead-mounted camera in his 1945 address; many realizations of such a camera are now available, including GoPro and Looxcie. Because these digital memories capture a large amount of data, it is hard to pinpoint interesting data. Numerous researchers have worked to address this challenge. However, to the best of our knowledge, we are the first in identifying unusual days by mobility data. *Haystack* from MIT [Adar et al. 1999] observes how a user interacts with information by recording what the user accesses. *Haystack* mostly focuses on information retrieval within a user's desktop environment, while our system focuses on help retrieving unusual data from a user's record of everyday life. Lamming and Newman [1992] from Xerox proposed using the type of activities in which the user was engaged when the data was stored to help information retrieval. However, they pointed out that necessary activity-sensing technologies were largely unavailable at that time. *Stuff I've Seen* from Microsoft [Dumais et al. 2003] provides a unified index of information regardless of the type of information (e-mail, web page, document, etc.) to allow users to easily find information they have seen before. Similar to *Haystack*, this system also focuses on the computer desktop environment.

Another application domain that can benefit from our system is healthcare [Frost and Smith 2003; Wilson and Atkeson 2004]. Healthcare monitoring systems can continuously monitor patients. While some monitoring systems focus on generating warnings, others are used for long-term diagnostic purposes. Since it is likely to be inconvenient and inaccurate for patients to record their activities or behaviors, these systems automatically record patients' activities through various sensors. These systems potentially record a large amount of data, and it thus becomes hard to retrieve or pinpoint important information. One way to apply our system is to use the user's mobility patterns and extract significant changes in the patterns. These changes may help identify causes of changes in a patient's medical conditions. For example, changes in sleeping patterns may indicate potential medical problems.

Our system may also help with mobility prediction [Song et al. 2006]. Predicting a user's mobility is important support for pervasive applications (such as a mobile Voice over IP) and for some context-aware systems. Prediction systems often build a profile using the history of user mobility patterns. If a user's mobility pattern changes significantly at a certain point in time, these predictors should dynamically adjust profiles. Our system can detect these changes, which can then be reported to the mobility predictors.

Section 2 describes wireless-network traces, and Section 3 presents our detection system. Section 4 presents the evaluation results of our system using wireless traces. Section 5 introduces a distillation application for digital memories that illustrates how our system can be used to decide an appropriate distillation level. Section 6 summarizes

Table I. Syslog events.

Month	Cisco	Vocera
April	8,693	71,713
May	11,440	114,899
June	7,059	156,743
Total	27,192	343,355

Table II. Number of user-days.

	Cisco	Vocera	Total
User-days	883	1,414	2,297
Users	29	95	124

our finding and suggests future work.

2. WIRELESS NETWORK TRACES

We used traces collected at wireless access points (APs) on the Dartmouth College campus. These syslogtraces consisted of MAC addresses of wireless devices that were associated with each access point, collected at the granularity of seconds. Along with these traces, we used the geographical location of access points to consider the proximity among access points.

We used the traces collected during three full months: April, May, and June 2004 [Kotz et al. 2005]. Based on Dartmouth's academic calendar, the months of April and May made up most of the spring term without any breaks. Spring term ended on June 8 and summer term did not start until June 24.

On the Dartmouth campus, there were two types of wireless network devices: on-and-off and always-on devices. The former mostly consisted of laptops, and the latter consisted of Cisco Voice-Over-IP (VoIP) wireless mobile phones and Vocera devices, which were a smaller version of VoIP wireless phones with speech-recognition capability. In this study, we focused on the always-on devices to have a manageable sized data set: 27,192 events were generated by Cisco phones and 343,355 events were by Vocera phones. Table I shows the number of syslog events recorded for each month. However, our mobility analysis can also be used for the on-and-off devices without any modification.

Although a wireless network user can carry more than one device with multiple wireless network cards or can carry different devices from day to day, we assume that a MAC address represents one user of a wireless device. We define a *user-day trace* to be a trace collected for one day (24 hours starting from 4 AM[1]) for a particular user. During the three months, we observed 2,297 unique user-days and 124 unique users as shown in Table II. Note that the number of user-days (2,297) is much smaller than the potential maximum value (11,284) if all users were active every day during the

[1]The campus-wide syslog traces roll over at 4 AM every day.

three months. This implies that most of these users did not regularly connect these devices to the network.

3. TRACE ANALYSIS MECHANISM

In this section, we describe our mechanism for trace analysis. The goal of our approach was to identify days that were different from the user's regular mobility pattern. For each day of the trace data, we computed the duration of stay at each location (e.g., building) the user visited during that day. If this *duration trace* matched any of the existing profiles of that user, we assigned the class of that matching profile to this user-day, and updated the profile with this user-day trace. Otherwise, we created a new profile using this user-day trace as the initial values. We describe each step in the following sections.

3.1 Aggregating Time

The density of APs on the Dartmouth College campus is high. The size of the campus is approximately 1 km^2, and we observed 503 APs in the three-month traces under this study. Due to the high density, many of the APs were closely located and a wireless user visiting the same location may have associated with different APs during each visit. Even during one visit, a user's device may have re-associated with multiple available APs in sequence. (This phenomenon is an instance of the ping-pong effect [Kim and Kotz 2007].) To discount these random events, we focused on the *total* duration of a user's visits at each AP during each day, instead of the duration of each visit. We acknowledge that by using the total duration rather than a sequence of durations, we omitted some information. However, this was inevitable due to noise in the traces.

Figure 1 shows the cumulative distribution function (CDF) of the duration of stay across all daily durations for three months. The solid line shows durations at APs. For

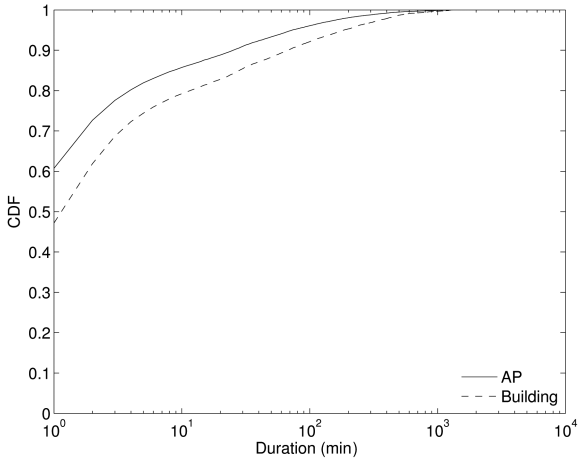


Figure 1. Cumulative distribution function (CDF) of daily durations at access points (APs) and at buildings across all durations.

the duration of APs, the 75th percentile is about 2-3 minutes. The percentage of short durations is high because a user associates with numerous APs while moving. Another reason is due to the ping-pong effect, which causes short-duration visits even when users are stationary; a device may associate with the closest AP for the most time, but briefly change to other nearby APs depending on the connectivity conditions

3.2 Aggregating Locations

As mentioned earlier, our syslog trace lists APs that each user visited. At a given location, a user can be associated with one of many available APs, and may rapidly switch APs even while stationary due to changes in the radio environment. Thus, we believe that the set of APs that the user visited is too fine-grained to represent a user's location history. Instead of using the actual set of APs, we aggregated APs into the buildings within which they were located. This aggregation effectively reduced the number of *user locations* from 503 APs to 126 buildings. After aggregation, each user-day trace contained a list of buildings with the total duration of stay at each building. Yoon et al. [2006] also uses this aggregation step in developing a realistic mobility model using the Dartmouth trace. As introduced earlier, Figure 1 shows the CDF across all durations. The dashed line shows the durations at building granularity. The 75th percentile is about 5-6 minutes. Not surprisingly, the durations at the building level are longer than those at the AP level.

3.3 Defining Regular Patterns Using Dynamic Profiling

After the aggregation steps, each user-day trace consists of a list of <building, duration> pairs. Given this trace, we need to create a profile that represents a user's regular mobility pattern. The profile can be computed as the average of individual user days. However, a user's regular pattern may change over time. For example, on a college campus, the mobility pattern is likely to change between academic terms and summer vacation. Without extra knowledge such as an academic calendar, it may be hard to generate an accurate profile for each time period.

In our approach, we did not use any extra knowledge. Instead, we dynamically updated a profile of each user. We used an exponentially-weighted moving average (EWMA) filter to update a user's profile. The widely-used EWMA filter generates a moving average, while smoothing out noise. At time $t + 1$, we updated each duration at building b in the profile, $e_{t,b}$, with the duration at the same building in the user-day trace, $d_{t,b}$, using an EWMA filter:

$$e_{t+1,b} = \alpha e_{t,b} + (1-\alpha) d_{t,b} \quad (1)$$

where α is the gain of the filter. We used an α of 0.5. We left the exploration of different values of α as future work.

Another interesting aspect of our approach is that we kept multiple profiles for each user. If a new user-day trace matched an existing profile, we updated the profile with the user-day trace. (We describe how we perform matching below.) If a new user-day trace did not match any of the existing profiles of that user, then we started a new profile. Each profile represented a class of that user's mobility.

3.4 Detecting Unusual Patterns Using a Pearson's Test

Given a user profile and a new user-day trace, we needed to decide whether or not the trace matched the profile. We considered several approaches. One popular test to decide whether two data sets were different was the t-Test. However, because this test compared the means between two distributions, it was not appropriate for our problem because our data consisted of paired sets, each data point consisting of a <building, duration> pair. Another popular test that compares two sets is the Kolmogorov-Smirnov test [Massey 1951]. However, this test compares the cumulative fraction plots of two sets and thus cannot be used for paired data lists.

The Pearson's correlation test could be applied to our data set. To perform the Pearson's test, two vectors under consideration need to be of the same length. Because we had 126 buildings, each user-day trace was converted to a vector of 126 elements with each element representing the duration at the particular building. Although this test assumes that the data follows a normal distribution, this assumption is not as strict as the data set becomes large. (Since our data set is large, we applied this test without checking whether or not our data in fact followed a normal distribution.) Given two variables x and y, Pearson's correlation coefficient is defined as:

$$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)\left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \tag{2}$$

where n is the number of data values. In our analysis, we only included the entries of those buildings whose values were non-zero in at least one of two vectors: user-day and profile vectors. Thus, n was defined as the maximum of the number of nonzero durations in the user-day trace and that in the profile. More precisely, n was the size of the set $\{\forall i | d_i \neq 0 \vee p_i \neq 0\}$ where d_i was the ith value of the user-day vector and p_i was the ith value of the profile vector. Then, the degrees of freedom were defined as $n - 2$. This value represented the number of independent data observations. We then compared r with the critical value (from the probability table) for the given degree of freedom using 95% confidence. If r exceeded the critical value, it meant that there was a statistically significant relationship between the user-day and profile. If n was either 1 or 2, we could not perform Pearson's test since the critical values were undefined for degrees of −1 and 0. For these undefined cases, we simply checked whether or not the locations in the profile and those in user-day trace were the same, without considering the duration.

4. EVALUATION

In this section, we present the results of our detection system using the Dartmouth wireless trace as the location data. We list mobility observations we made from this analysis, illustrate how our system classifies user-days, and summarize its capabilities.

4.1 Classification Result

In our trace, we have 124 VoIP users. While some users are regular users who were

active most days, the rest did not use their device regularly. In our analysis, we focused on the regular users, defined as users who connected to any access point more than seven days during the three-month period. We found 64 users (52%) to be regular. Figure 2 shows the CDF of the number of active days across 124 users.

We applied our classification mechanism to 64 regular users. For each user, our system classified only the active user days. Recall that our system created a class whenever a new user-day trace could not be classified into one of the existing classes. We called any class that contained five or more user-days a baseclass. Figure 3 shows the histogram of users with a different number of base classes. 71.9% of users had only one base class during the three month period, while 23.4% had more than one base class.

We first needed to understand the percentage of user days that could be considered as regular versus unusual. Figure 4 shows the CDF across all user days. The x-axis

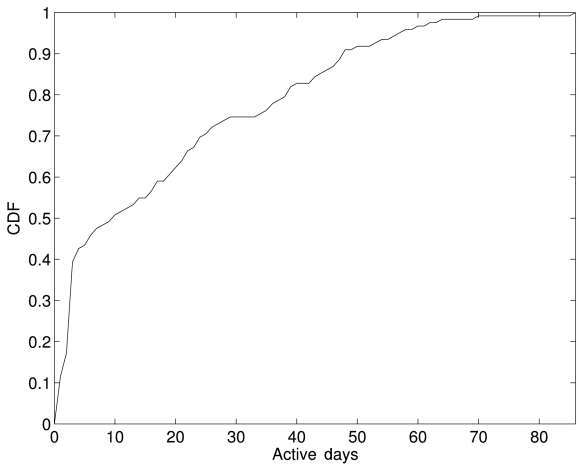


Figure 2. Cumulative distribution function (CDF) of users with the specific number of active days across 124 users. 52% of users were active more than seven days.

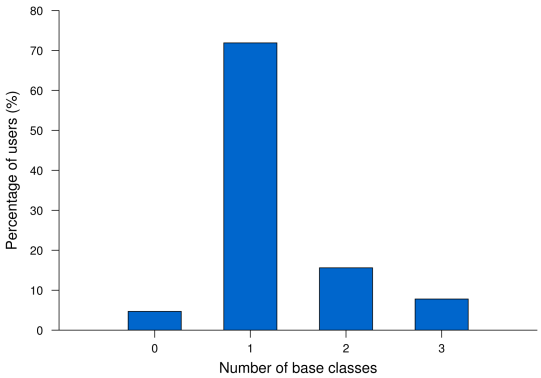


Figure 3. Histogram of users with a different number of base classes.

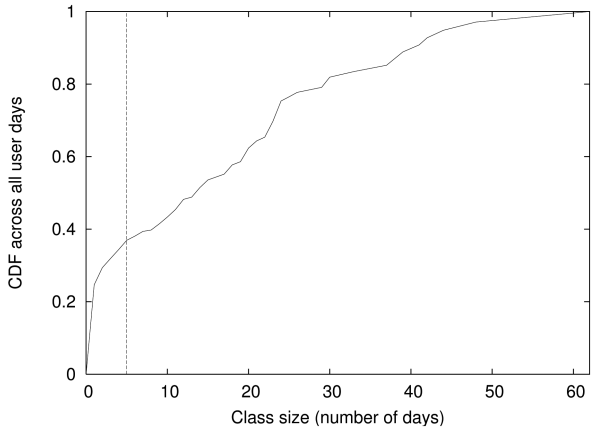


Figure 4. Cumulative distribution function (CDF) of classes with specific size (in terms of days) across 2128 user days. The default threshold to be considered as a base class is 5 days.

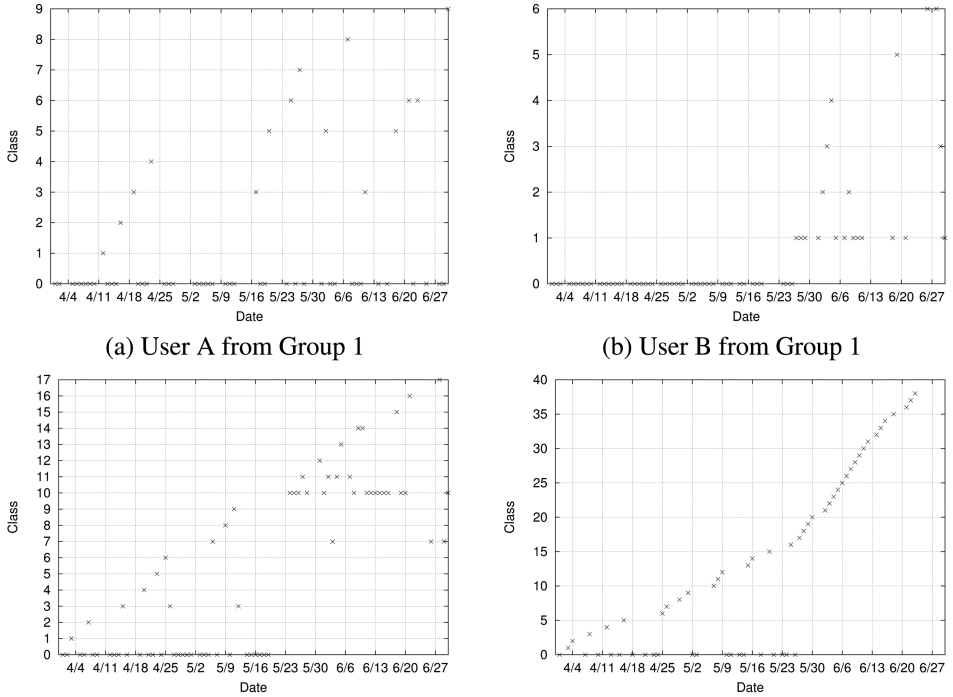
Table III. Percentage of unusual days.

Group	r_b	p_u	Number of users (%)	Average active days
1	$r_b \geq 1.5$ or $d_u = 0$	$p_u \leq 40\%$	40 (62.5)	36.2
2	$0.5 \leq r_b < 1.5$	$40\% < p_u \leq 67\%$	16 (25.0)	28.9
3	$r_b < 0.5$	$67\% < p_u \leq 100\%$	8 (12.5)	27.1

shows the class size in terms of days and the y-axis shows the cumulative percentage of days that belong to a class with a specific size. First, using our threshold of 5 days to be considered as a base class, we found that 34% (731 days) of user days were considered to be unusual. If we reduced the threshold, the percentage of unusual days would reduce. We chose the default threshold to be a little smaller than the cut off for identifying the active users. Since we focused on the users who were active more than 7 days, we chose our threshold to be 5 days. Second, the majority of unusual days were a one-time event. In other words, the mobility pattern did not repeat. Among all the unusual days, 72% belonged to a class of size 1 (day).

We then considered how many of the active users had a regular mobility pattern. To quantify this, we defined r_b as the ratio of the number of days that belonged to base classes (d_b) to the number of days that did not belong to base classes (d_u). Then, the percentage of dates that were considered to be unusual was computed as $p_u = d_u / (d_b + d_u)$.

Table III shows r_b and pu. The first group ($r_b \geq 1.5$) of users showed a regular mobility pattern. Sixty percent or more of their days belonged to a base class, while the rest of the days were considered to be unusual. The second group ($0.5 \leq \mathrm{rb} < 1.5$) included 25% of users. This group of users had one or more base classes, but roughly half of the days do not belong to a base class. The third group of users either did not have any base class or the majority of their days did not belong to a base class. The last column of this table shows the average number of active days for the users in each



(a) User A from Group 1 (b) User B from Group 1

(c) User C from Group 2 (d) User D from Group 3

Figure 5. Sample users. The x-axis is the calendar date and the y-axis is the class into which the mobility pattern of the particular day falls. Note that the dates labeled on the x-axis (and drawn with grid lines) denote Sundays.

group. Group 1, whose days mostly belonged to a base class, had the largest number of active days (36.2 days). The other two groups had a smaller number of active days than that of Group 1, but they were still large enough to have a base class (5 days).

Figure 5 shows the class assignment across three months for several sample users extracted from each of the three groups of Table III. The x-axis shows time and the y-axis shows the class into which the mobility pattern of the particular day falls. Note that the dates labeled on the x-axis (and drawn with grid lines) denote Sundays. User A had only one base class (Class 0) and the majority of his or her days belonged to this class. User B had two base classes (Class 0 and 1) and again a majority belonged to these base classes. In User C's case, roughly half belonged to base classes. A majority of User D's days did not belong to a base class.

Through this study, we also discovered several interesting mobility characteristics. First, we did not observe any repeating pattern based on the day of the week. The only weekly pattern that we observed was that most users were active during only the week days and not active during weekends (User A and B in Figure 5). Second, a change in mobility patterns coincided with an academic calendar. For a set of users, their mobility pattern was regular during academic terms but became irregular during summer vacation. For the first two months, these users had a base class and

Table IV. Additional mobility characteristics.

Characteristic	Number of users (%)
Switch from one base class to irregular	6 (9.4)
Switch from one base class to another	7 (10.9)
Switch among multiple base classes	8 (12.5)

most of their days belonged to this base class. Then, toward the end of May, which is when the term ended for that academic year at Dartmouth, these users' days did not belong to any base class (User D). For another set of users, their mobility pattern shifted from one base class to another towards the end of May (User C). Third, while most users had one base class, some users had multiple base classes, often switching back and forth among them.

The mobility characteristics are important to understand for pervasive application developers during their design and testing. For example, to test new software in a simulation environment, they need to know how many users follow certain characteristics. Table IV shows the number of users that followed the mobility characteristics described above. Note that the first observation applied to most users and thus we did not include it in this table.

In summary, we identified unusual days using Pearson's test. If r did not exceed the critical value, then the corresponding day was considered to be different from usual days (which belonged to one of base classes). However, for those users who did not show any regular mobility patterns, we could not identify any unusual days since there were no regular or usual days.

4.2 Users on the Map

In the previous section, we identified many interesting mobility characteristics of individual users using our system. In this section, we look into some unusual days that we identified in the previous section to deepen our understanding of the findings.

We first considered five active days of User A in detail: 5/28, 6/1, 6/2, 6/3, and 6/4. Among these five days, only 6/2 was identified as a different class (See the graph in Figure 5). Table V depicts the locations of buildings (on an invisible campus map) that User A visited on each day. Each circle shows a building and the filled circles show

Table V. Buildings User A visited during five days. Circles denote the location of all the buildings that the user visited on that day. The black circles represent those buildings where the user stayed longer than 30 minutes.

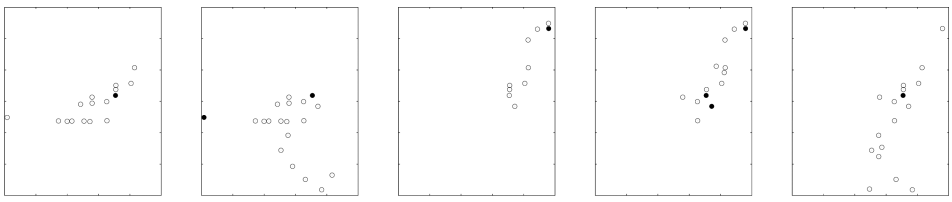
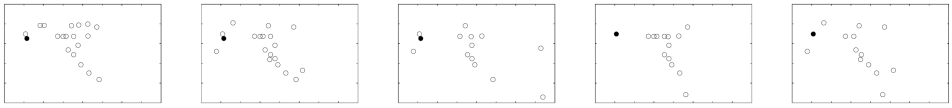


Table VI. User A: Total duration of stay at each building (in minutes). The table shows only the buildings where User A stayed longer than 30 minutes in any of the five days.

Dates	5/28	6/1	6/2	6/3	6/4
Building 1	**53**	**125**	12	**91**	**100**
Building 2	13	**140**	0	0	0
Building 3	0	0	**183**	**99**	1
Building 4	0	2	3	**36**	3
Class	0	0	5	0	0

Table VII. Buildings User B visited in five days. Circles denote the location of buildings that the user visited on that day. The black circles represent those buildings where the user stayed longer than 20 minutes.



5/24 5/25 5/26 5/27 5/28

those buildings where the user stayed longer than 30 minutes. Note that our system used both the location and time information, while this figure shows only the location without the exact duration of stays. For a better understanding of why 6/2 was classified into a different class, we also showed the duration of stay at buildings (where the user stayed for more than 30 minutes total) in Table VI. From Table VI, we can clearly see that User A spent much more time at Building 1 on those four days (in Class 0) while she did not visit Building 1 and stayed mostly in Building 3 on 6/2 (in Class 5).

Another case that we considered was class transition. User B in Figure 5 shows a class transition during the week of 5/23. Days up to 5/26 belong to Class 0 and the days after that belong to Class 1. Table VII shows the buildings that User B visited in five days: 5/24, 5/25, 5/26, 5/27, and 5/28. The circles show the location of the buildings and filled circles show the buildings where users spent more than 20 minutes. Table VIII shows the duration of stay at the two buildings. It is clear how the days are classified into two different classes. User B stayed at Building 5 in the beginning of the trace and then stayed mostly in Building 6.

In summary, as illustrated with sample users, we could easily do the following using our system:

Table VIII. User B: Total duration of stay at each building (in minutes). The table shows only those buildings where User B stayed longer than 20 minutes in any of the five days.

Dates	5/24	5/25	5/26	5/27	5/28
Building 5	**364**	**28**	**21**	0	0
Building 6	7	0	0	**31**	**86**
Class	0	0	0	1	1

- Identify days when mobility patterns change.
- Identify days where the pattern looks like that of many other days, i.e., a typical day.
- Identify days where the pattern does not look like (many) other days, i.e., an unusual day.
- Discover the general characteristics of human mobility.

5. SAMPLE APPLICATION SCENARIO

In this section, we introduce a sample pervasive application and how our mechanism interacts with this application.

Digital memories or diaries often use a body-mounted camera to capture photos or movie streams. Because this data can quickly accumulate, it is important to distill or compress the data. One simple way to distill the data would be to filter out data frames or photos at fixed intervals. However, this static method of distillation would lose many details that the user may later find interesting. Perhaps distillation should be done differently if a certain day was special or interesting. For many people whose jobs did not vary much from day to day, the system probably would not record many new things every day. A typical day for such people would be images of home, the route to work, and the office. However, when a person takes a vacation and travels, the camera is likely to capture many new images. Thus, the distillation system should capture these special days and apply different filters to the data according to type of days that the data was gathered.

We now consider this application in more detail and how it interacts with our system. The head-mounted camera captures images during the day. Locations of the user are also recorded. At the end of the day, the user backs up the data to a storage server. For a fixed time period, the server keeps the original data. When the data is about to expire, the server needs to decide how to distill the data. To make the decision, the server sends location information to our system, and our system determines the significance of that day. The significance level may be either determined using a system default (e.g., highest level for the first day of a class) or predefined by the user. Our system returns the significance level to the server and the distillation server compresses data according to the significance level.

Although our current system detects unusual days solely based on mobility patterns, we recognize that we ultimately need to combine other contextual information to detect unusual days. For example, an office worker may spend the usual amount of time in the usual places, but something about their work that day (such as the people they met, conversations they had, or tasks they performed) may have been unusual. We can adapt other sensor-based systems to collect different categories of information. For example, activity recognition systems [Consolvo et al. 2008, Choudhury et al. 2008] may provide a series of activity data, and this data can be put into our detection system to identify unusual activities. Using both activities and mobility patterns, we can identify unusual days that may be more meaningful to users.

6. CONCLUSIONS

Numerous pervasive applications such as digital memories collect a large amount of data now that sensing devices and storage are both cheap and readily available. One

key challenge is to identify and retrieve interesting data. In this paper, we presented a system that automatically identified unusual days using location information. We evaluated our detection system using real wireless-network traces collected on the Dartmouth campus. Our system identified regular mobility patterns, detected changes in the patterns, identified unusual days, and also extracted general characteristics of human mobility. Although we evaluated our system using wireless-network traces, we expect that other types of location information would also be easy to collect using devices such as GPS.

In the future, we would like to develop a distillation application for digital memories as described in Section 5, and use our system for indexing and adaptive distillation. We would then collect image data along with location information and use our system to effectively manage the vast amount of image data.

ACKNOWLEDGEMENT

We acknowledge the support of the Center for Mobile Computing, and the data sets provided by the CRAWDAD archive, at Dartmouth. We would like to thank Hyungjin Myra Kim at the University of Michigan. With her deep knowledge on statistics, she provided insightful comments on the statistical issues throughout the design process of our system.

REFERENCES

ADAR, E., KARGER, D., AND STEIN, L. A. 1999. Haystack: per-user information environments. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, 413-422.

BUSH, V. 1945. As we may think. *The Atlantic Monthly 176*, 1, 101-108.

CHOUDHURY, T., BORRIELLO, G., CONSOLVO, S., HAEHNEL, D., HARRISON, B., HEMINGWAY, B., HIGHTOWER, J., KLASNJA, P., KOSCHER, K., LAMARCA, A., LANDAY, J. A., LEGRAND, L., LESTER, J., RAHIMI, A., REA, A., AND WYATT, D. 2008. The mobile sensing platform: an embedded activity recognition system. *IEEE Pervasive Computing 7*, 2, 32-41.

CONSOLVO, S., MCDONALD, D. W., TOSCOS, T., CHEN, M. Y., FROEHLICH, J., HARRISON, B., KLASNJA, P., LAMAREA, A., LEGRAND, L., LIBBY, R., SMITH, I., AND LANDAY, J. A. 2008. Activity sensing in the wild: a field trial of UbiFit Garden. In *26th Annual CHI Conference on Human Factors in Computing Systems,* 1797-1806.

CZERWINSKI, M., GAGE, D. W., GEMMELL, J., MARSHALL, C. C., PEREZ-QUINONESIS, M. A., SKEELS, M. M., AND CATARCI, T. 2006. Digital memories in an era of ubiquitous computing and abundant storage. *Communications of the ACM 49*, 1, 45-50.

DUMAIS, S., CUTRELL, E., CADIZ, J. J., JANCKE, G., SARIN, R., AND BOBBINS, D. C. 2003. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 72-79.

FROST, J. AND SMITH, B. K. 2003. Picture of health: photography use in diabetes self-care. In *Proceedings of UbiComp 2003, the Fifth International Conference on Ubiquitous Computing*.

KIM, M. AND KOTZ, D. 2007. Periodic properties of user mobility and access-point popularity. *Personal and Ubiquitous Computing 11*, 6, 465-479.

KOTZ, D., HENDERSON, T., ABYZOV, I., AND YEO, J. 2005. CRAWDAD metadata: dartmouth/campus/movement (v. 2005-03-08). http://crawdad.cs.dartmouth.edu/meta.php?name=dartmouth/campus/movement.

LAMMING, M. G. AND NEWMAN, W. M. 1992. Activity-based information retrieval: technology in support of personal memory. In *Proceedings of the IFIP 12th World Computer Congresson on Personal Computers and Intelligent Systems--Information Processing '92 Vol. 3*, 68-81.

MASSEY, F. J., JR. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association 46*, 253, 68-78.

SONG, L., DESHPANDE, U., KOZAT, U. C., KOTZ, D., AND JAIN, R. 2006. Predictability of WLAN mobility and its effects on bandwidth provisioning. In *25th IEEE International Conference on Computer Communications* 2006.

WILSON, D. H. AND ATKESON, C. 2004. Automatic health monitoring using anonymous, binary sensors. In *CHI Workshop on Keeping Elders Connected*, 1719-1720.

YOON, J., NOBLE, B. D., LIU, M., AND KIM, M. 2006. Building realistic mobility models from coarsegrained traces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, 177-190.

Minkyong Kim is a research scientist at IBM T.J. Watson Research Center in New York. Dr. Kim is working in the area of messaging systems and cloud computing. More broadly, her research interests include distributed systems, mobile computing and pervasive computing. She received her Ph.D. in Computer Science and Engineering from the University of Michigan in 2004 and worked as a postdoctoral research fellow at Dartmouth College for two years, prior to joining IBM in 2006. She received her B.S. and M.S. in Computer Engineering from Seoul National University. She published numerous papers at journals and conferences including INFOCOM, MobiCom, MobiSys, Pervasive, FAST, ICDCS, ICNP and Middleware.

David Kotz is the Champion International Professor, in the Department of Computer Science, and Associate Dean of the Faculty for the Sciences, at Dartmouth College in Hanover NH. During the 2008-09 academic year he was a Visiting Professor at the Indian Institute of Science, in Bangalore India, and a Fulbright Research Scholar to India. At Dartmouth, he was the Executive Director of the Institute for Security Technology Studies from 2004-07. His research interests include security and privacy, pervasive computing for healthcare, and wireless networks. He has published over 100 refereed journal and conference papers. He is an IEEE Fellow, a Senior Member of the ACM, a member of the USENIX Association, and a member of Phi Beta Kappa.

After receiving his A.B. in Computer Science and Physics from Dartmouth in 1986, he completed his Ph.D in Computer Science from Duke University in 1991 and returned to Dartmouth to join the faculty. For more information see http://www.cs.dartmouth.edu/~dfk/.