Prediction for Mobile Application Usage Patterns *

Chang Tan[1], Qi Liu[1], Enhong Chen[1], Hui Xiong[2]
{tanchang, feiniaol}@mail.ustc.edu.cn, cheneh@ustc.edu.cn, hxiong@rutgers.edu
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]MSIS Department, Rutgers Business School, Rutgers University

ABSTRACT

Advances in smartphone technology have enabled the prevalence of mobile applications. Such a variety of mobile applications make the smartphone more interesting and more humanized, and running these applications has become the major function of smartphones. However, the limited resources of current smartphone requires both researchers and companies paying more attention to the way of effectively managing mobile applications. A challenging task is how to predict mobile user's application usage patterns for improving smartphone performance. Although the usage patterns can be treated as the time series prediction problem, traditional time series models are usually too complex to be directly adopted by the smartphone environment. Novel methods that can be running on smartphones, are desperately needed. To this end, in this paper we propose a simple and efficient method named Prediction Algorithm with Fixed Cycle Length (PAFCL), which can be used for predicting mobile application usage patterns in real-time and mobile environment. The experiments on the collected real-world data validate the ability of our prediction methods.

Keywords

mobile application, prediction, usage pattern

1. INTRODUCTION

Recent years, mobile applications get rapid development due to the advances of the smartphone technology. Now the smartphones usually support various applications and services beyond the traditional speech-centric service, such as music, videos, web browsing, gaming and camera shooting, just to name a few. Consequently, a large number of mobile applications are downloaded and installed, and more importantly, it is common for a smartphone user to open or run multiple applications simultaneously. However, the limited capacity of both battery and memory becomes the bottleneck of the smartphone performance, because several applications running simultaneously will consume a lot of resource, and some of these consumptions are even unnecessarily. Furthermore, too many running applications will also increase reaction time and affect the mobile user experience.

Thus, it is critical for effectively managing the mobile applications so as to improve the smartphone's performance. Meanwhile, it is impossible to let user do this kind of job by themselves, since this will require a lot of user interventions. Alternatively, if we profile users based on their usage patterns, we can leverage these profiles for developing automatical application managing techniques.

Therefore, in this paper we focus on automatically determining and predicting each user's frequently-used applications in a fixed time slot(e.g. one hour), i.e., finding usage patterns. More importantly, to protect user privacy and save the data flow, it is desirable to directly perform usage patterns mining on users' smartphones instead of transiting the raw context data to the back end server and then performing mining. In this way, the smartphone will efficiently know which application the user may use and respond even before user requests. For example, when a user wants to send a message, his smartphone guessed the motivation and showed the message application on screen, then user can write the message directly without finding and opening the application manually. After sending the message, the smartphone estimates he will not use the application again for a long time, and it can shut down the application automatically, both saving the energy cost and freeing memory for upcoming applications. This process really makes the smartphone more "smart" and energy-saving, and meanwhile improves user experiences.

From the mobile application usage logs, the usage time series can be generated expediently. Forecasting these usage time series is a possible way to judge whether or not an application shall be used [2]. However, the problem of predicting mobile application usage patterns cannot be addressed by traditional time series models [2, 1] since these models are usually too complex and time consuming to be adopted by smartphone for making real-time prediction. Thus, novel prediction methods that are both simple and effective, are desperately required.

To this end, we explore the unique characters of the mobile application usage logs. Specifically, we note that there are some periodic changes in mobile user behavior. It is common that most people are working and resting by the rule of 24

*This material was prepared for the Mobile Data Challenge 2012 (by Nokia) Workshop; June 18-19, 2012; Newcastle, UK. The copyright belongs to the authors of this paper.

Nokia MDC Workshop '2012 Newcastle, UK

hours per day and 7 days per week. Meanwhile, it is easy to understand that recently used applications may be used again sooner or later. Thus, we consider these two factor to develop novel method. Along this line, we propose a simple but efficient prediction method named Prediction Algorithm with Fixed Cycle Length (PAFCL), which considers both the periodic changes of behaviors and the influence of recent actions of mobile users. Experimental results on real-world data which is collected by Nokia show that our methods can predict usage patterns more precisely.

2. PROBLEM STATEMENT

To ease presenting the problem of predicting mobile application usage patterns, we first define several notations.

Definition 1. *The mobile application usage pattern is the mobile application usage habits of a specific user in special time slot.*

By analyzing the past mobile application usage patterns hiding in the usage logs, we can predict the future usage patterns with some methods. Since we focus on the methods that are suitable for running on each smartphone, the following discussion just depends on the usage logs of the specific user. As have said, the application usage logs mainly contain when and which the application is used by the specific mobile user. Segmenting the logs into the equal time slot, the time series of application usage numbers can be generated.

Definition 2. *The time series* $N_A = \{N_{A1}, \ldots, N_{At}\}$ *presents the usage number* N_{Ai} *of mobile application* A *by specific user in each time slot* T_i, $i = 1, \ldots, t$.

Given specific user and the length of the time slot, we can generate time series for each mobile application that the user has used. In other words, each time series stands for the way of the user using a mobile application in a specific time slot. It is obvious that, given a time series N_A from T_1 to T_t, the next value $N_{A,t+1}$ can be predicted. Correspondingly, the prediction result can be considered as the possible usage number for each application in the next period. Similarly, by predicting the value $N_{A_j,t+1}$ for each mobile application A_j, the usage pattern of the specific user in time slot T_{t+1} can be generated. In this way, we can achieve the target that predicting mobile application usage pattern of each mobile user. Based on the prediction result, it is easy to determine that the application with higher predicted number should be used in the next time slot.

Definition 3. *For each used application* A_j *of one mobile user, there is time series* N_{A_j} *from* T_1 *to* T_t. *Given the time slot* T_{t+1}, *the new value* $N_{A_j,t+1}$ *can be forecasted. Comparing these values, we can predict which application will be used with high probability in the time slot* T_{t+1}.

Using the prediction results, the smartphone can effectively manage the mobile applications to improve its performance and enhance user experience, and meanwhile avoid the wastage of both energy and memory.

3. PREDICTION ALGORITHM

Though predicting the mobile application usage pattern can be defined like a time series prediction problem, there

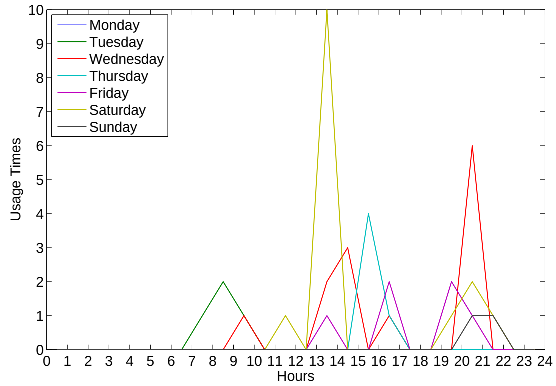

Figure 1: App "Text Message" usage statistics of one user in a week.

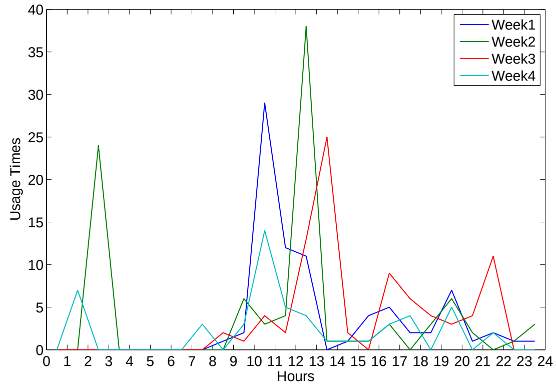

Figure 2: App "Text Message" usage statistics of one user in consecutive four weeks.

are still some difficulties. As we all know, most traditional time series models, such as ARMA[1], are complex and supervised. Which means before we determine that a traditional time series model can forecast a time series correctly, we should test many groups of parameters. Furthermore, when we consider these models for the smartphone and seek to predict usage patterns for each user, there are some other problems. At first, although the smartphone becomes more and more intelligence and powerful, the capacity of both battery and memory still limit its performance. Thus, if we build and evaluate time series models for each application on the smartphone, this process may cost the great mass of energy and memory. This could lead to bad user experiences, and it is also against our propose for saving energy and memory. Secondly, preparing and training groups of parameters for each different user is a very difficult problem. Meanwhile, pre-prepared parameters may not be suitable for user's changing demands, and a self-learning model should be a better choice than the traditional supervised models. In summary, the traditional methods which are complex and without self-learning usually consume too much energy and memory, and also lead to bad user experiences. Thus, we consider to propose both simple and efficient unsupervised method for mobile application usage pattern prediction.

Worth noting that, the mobile application usage data has its unique characters. For more reasonably exploring these unique characters, we overcome the complexity of the traditional models, and design a novel unsupervised prediction approach.

First, we note that there are some periodic changes in mobile user behavior. For example, the user may listen music in the daytime, browse the web in the night, and do not use any application when sleeping. Meanwhile, each user may use more message applications in weekdays while more game applications in the weekend. Thus, there are some periodicity in the mobile application usage (e.g., an application may be used more frequently in special time slot than others periodically). Figure 1 and 2 show the periodic changes of one anonymous real user with id "002". In Figure 1, we show the usage number of an application named "Text Message" used by him in one week, where the usage numbers are counted by per hour. It is clear that this user usually uses this application in two periods, one is in afternoon, and another is at night. Figure 2 shows an even longer periodic changes. Specifically, each value in this figure means the sum of usage times in specific hour of all the days in a week. Here, we also choose user "002" and application "Text Message" to demonstrate the periodicity. Similar to this example, the periodicity exists for most of users and their applications, and the ubiquity of the periodicity has been proved by the experimental results. Considering the periodic factors, we can formalize the original prediction problem to the periodic prediction problem. With the existence of periodicity, one usage time series can be now segmented into h subseries/cycles $N_{Ai} = \{N_{A,i}, N_{A,i+h}, \ldots, N_{A,i+kh}\}$, $i = 1, \ldots, h$. If the the cycle is obvious and credible, we just need to consider the subseries N_{Ai} for predicting the value $N_{A,i+(k+1)h}$, rather than the whole series. Meanwhile, for most of the mobile users, the cycle of one day (i.e., 24 hours) and one week (i.e., 7 days) are both obvious and credible. In this way, we segment the original time series by each cycle, and based on the segmentation we propose pertinent and simple methods.

Also, we notice that users' recent behavior are usually more important for the prediction process. The same application that is used one week before or just one day before will express different degree of importance. It is possible that the user runs one application in a time slot, and he may use it again in the next period. For instance, the user may play a game for some time, then stop it and after taking a rest he will continue to play it. During the short break, if we make predictions for this user, the specific game application should have high rank than other applications that were used before hours. Thus, both the historical effects and the recent behaviors should be considered by the prediction methods.

For considering these restricted conditions, we propose a prediction framework: Prediction Algorithm with Fixed Cycle Length (PAFCL). In PAFCL, there are two parameters, one is the time slot length l (e.g., 1 hour) used for segmenting the logs, another one is the cycle length h which can be fixed into smartphones by analyzing mobile users' general behaviors. For example, fixing 24 hours as the cycle length may be a good choice, because it is the work and rest routine for most users. With fixed l and h, the PAFCL can record usage logs using time series and predict usage patterns in real-time on the smartphone. When predicting which applications will be used in the time slot $T_{t+1} = T_{i+kh}$, $k = 1, 2, 3, \ldots$, for each application A_j the subseries $N_{A_j,i}$ will be chosen and the new value $N_{A_j,i+kh}$ can be computed based on $N_{A_j,i}$. Then the predicted values for all applications can be sorted by descending order. In this way, the posterior management can be conducted automatically.

It is clear that the key objective of PAFCL is to predict the new value $N_{A_j,i+kh}$. According to the framework of PAFCL, we use two methods to get the new value, one is Exponentially Weighted Moving Average (EWMA)[3], and another mainly use Cumulative Probability Distribution (CPD).

EWMA. Exponentially weighted moving average applies weighting factors and these factors decrease exponentially. Thus, the newer data points have higher influence for the weighting sum. We see the value $N_{A_j,i+kh}$ as the weighting sum S_k of EWMA for the subseries $N_{A_j,i}$, and $N_{A_j,i+kh}$ can be computed as follows:

$$N_{A_j,i+kh} = S_k = \begin{cases} N_{A_j,i} & k = 1 \\ \lambda N_{A_j,i+(k-1)h} + (1-\lambda) S_{k-1} & k > 1 \end{cases}$$

CPD. It is obvious that one application has different usage probabilities in different time slot of a cycle. The probability of application A_j used in the i-th slot can be computed by the usage times:

$$P(A_j, i) = \frac{\sum_{k=0}^{\lfloor (T-i)/h \rfloor} N_{A_j,i+kh}}{\sum_{t=1}^{T} N_{A_j,t}}$$

where $\sum_{k=0}^{\lfloor (T-i)/h \rfloor} N_{A_j,i+kh}$ means the sum of the usage numbers in each i-th slot, and $\sum_{t=1}^{T} N_{A_j,t}$ is the sum of the usage numbers of application A_j. Since the probabilities of each used application in the specific time slot are computed, we can choose applications with higher probability as the candidates. However, there still exists the bias problem for the occasional used applications. For example, if a mobile application is only used once by specific user for a very long time, its usage series may like this $\{0,\ldots,0,1,0,\ldots,0\}$, and the probability will be 100% in the used time period. That is really a high value, but meaningless for the prediction. Thus, we introduce a penalty factor to solve the problem:

$$P(A_j, i) = \frac{\sum_{k=0}^{\lfloor (T-i)/h \rfloor} N_{A_j,i+kh} - \lfloor (T-i)/h \rfloor}{\sum_{t=1}^{T} N_{A_j,t}}$$

where the penalty factor $\lfloor (T-i)/h \rfloor$ means the application should be used at least once per cycle on average, so that occasional used applications will have lower values than common used ones.

Complexity. Noting that both EWMA and CPD can be used for incremental learning because the S_k and $P(A_j, i)$ can be calculated recursively, the smartphone just needs to keep a $m \times h$ matrix in memory and then the new observed values can be easily used for the next computation. Each entry a_{jk} of the matrix is the current value of S_k or $\sum_{k=0}^{\lfloor (T-i)/h \rfloor} N_{A_j,i+kh}$ of application A_j in the k-th slot, m is the number of used applications. And the computation complexity of making prediction in a time slot is close to sorting the new value array. Thus, the PAFCL framework achieve the objectives of both low complexity and self-learning with few parameters.

4. EXPERIMENTS

We conduct extensive experiments on real logs of mobile users. These rich context data are collected by Nokia from 38 volunteers including application usage log, GPS data, system information, GSM data, call log, sensor data, etc. Each volunteer has an anonymous id, such as user "002" in Figure 1 and 2, and the range of the mobile logs length is

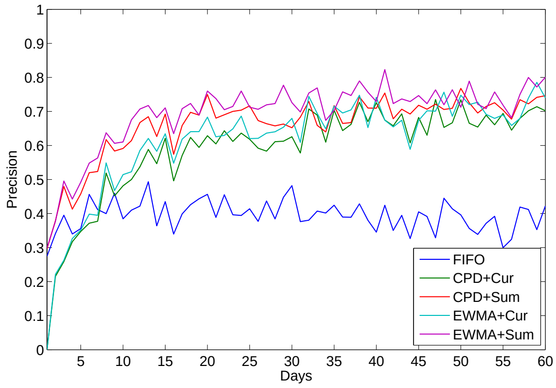

Figure 3: Average precision per day for all users in 60 days.

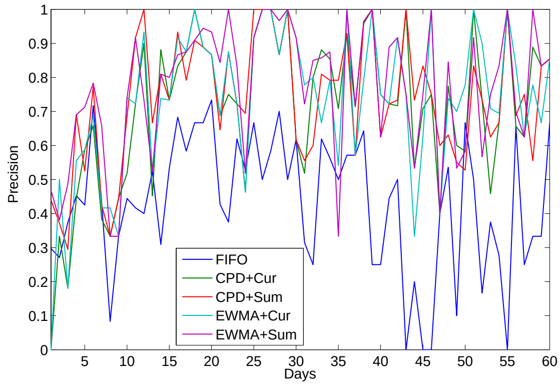

Figure 4: Average precision per day for user “077” in 60 days.

Table 1: Average Precision Comparison

Method	Precision
FIFO	0.395
CPD+Cur	0.591
CPD+Sum	0.660
EWMA+Cur	0.615
EWMA+Sum	**0.695**

from two months to one year. In this paper, we only consider the mobile application logs.

To evaluate the prediction ability of our methods under the framework of PAFCL, we choose *Precision* as the evaluation metric. For a given time slot t, we choose the top 5 predicted applications as the result set P_t of each method. We also have a real-used application set U_t for the given time from the usage logs, which serves as the ground truth. Comparing these two sets, the intersection of them are the correct predictions, and the precision in given slot t can be computed by: $Precision_t = |P_t \cap U_t|/|U_t|$. One step further, the average precision in T slots for a longer time range is: $Precision_T = \sum_{t=1}^{T} Precision_t/T$.

For each method, we choose the first 60 days' log of each user as the training set. Specifically, we segment the usage log by hour, i.e., the time slot length l is an hour, and similarly, the cycle length h is 24 hours. Besides the predictions made by EWMA and CPD, we also compare the results of considering the influence of neighbor time slots or not, i.e., using the sum of current hour and its neighbor hours' usage numbers as the prediction value (denoted as Sum) or just using the value of current hour (denoted as Cur). Meanwhile, as a baseline, we use a FIFO (First In First Out) queue simulating the real usage situation of smartphone memory. In order to be more real realistic, the FIFO queue will be cleared each day as the smartphone shuts down once per day. In all, there are five methods for evaluation, EWMA+Sum, EWMA+Cur, CPD+Sum, CPD+Cur, FIFO. Among them, the first four are our proposed methods, under the framework of PAFCL.

Figure 3 illustrates the average precision per day for all users in 60 days. It clearly shows that EWMA+Sum and CPD+Sum are more precise than other methods because they consider the influence from all the factors. The EWMA+Cur and CPD+Cur get zero at the first day, because it is the first time slot and all the subseries are starting with zero, and thus no result can be predicted. All the four methods based on PAFCL has the ability of self-learning as observed from their rising curves, while in contrast, FIFO almost keeps a horizontal line. We also calculate the total average precision of all the 60 days for all the users and the results are shown in Table 1, where EWMA+Sum and CPD+Sum also get the better results than others, and EWMA+Sum is the best among all the methods.

As an example, Figure 4 shows the prediction results for user “077”. It shows that each prediction method's precision result varies, and this may because some factors still have not been considered in our methods. We also find that EWMA+Sum and CPD+Sum get better results in most days, and all the methods based on PAFCL outperform the baseline FIFO method.

5. CONCLUSION

In this paper, we propose a simple but efficient framework named Prediction Algorithm with Fixed Cycle Length (PAFCL), which can be used on the smartphones for real-time predicting each user's mobile application usage patterns. The PACFL considers both the periodic changes of user behaviors and the influence from mobile user's recent actions, and thus achieve the objective of both low complexity and self-learning with few parameters. Experimental results on the real-world data which is collected by Nokia show the effectiveness of PAFCL, and the proposed methods under this framework usually predict usage patterns more precisely.

6. REFERENCES

[1] G. Gardner, A. Harvey, and G. Phillips. An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of kalman filtering. *Journal of the Royal Statistical Society*, 29(3):311–322, 1980.

[2] J. Hamilton. *Time series analysis*. Cambridge Univ Press, 1994.

[3] J. Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210, 1986.