

The Machine Learning Solution based on Period and Deep Construction of Mobile Data for Predicting User Habit

Juan Du

New Research and Development Center of Hisense, Qingdao 266071, China

dqxwpl@sina.com

Abstract. Information analysis on user habit is becoming hotter as its value for many potential and profitable areas. However, how to excavate hard-core data and obtain the most effective information associated with research target, and how to obtain more precise prediction of user habit timely and fast on mobile terminals, are still big challenges. Most of the traditional methods use static data, which can no longer meet the requirements of the movable era, such as learning daily travel route and predicting the next most possible applications. Additionally, though algorithms based on Artificial Intelligence (AI) have boomed, many researches only based on direct data and pay insufficient attention to the deeper interrelation of data. This paper introduces two relevant AI models improving the phone memories; then, the main solution is recommended in detail. It mines some closely correlated parameters by specific mechanism named Linked Trigger (LT) and filtering policies for positioning, digging their underlying relations for better learning user habit by special construction 'Directed Location Pair Application (DLPA)'. Based these analysis on the deep-seated connection between the collected data, the conditional probability referring Bayesian network is used to learn and predict the location habit and application habit.

1. Introduction

In the course of designing the mobile terminals, there are many problems closely related with the lack of user habit learning. One of the most serious industrial difficulties is on the memory management. For one thing, Linux system uses Least Recently Used (LRU) mechanism to kill the least frequently used application for saving memory, which means those applications that are not used frequently enough but would probably be opened in the next minute, may possibly be killed by the LRU policy before it is started according to user's recent habit. For another thing, Android system applies the method named Low Memory Killer (LMK) to manage the memory. LMK kills the applications with relatively lower priority to tackle with the memory intension. This follows that the application whose priorities are not high enough may be killed by LMK, even though they are likely to be used soon referring to the habit of the user. Obviously, both of these two systematic memory management policies lead to same problem: when users try to start up the applications again according to their latest usual habit, phone may have already lost the latest status data of the application, so they have to restart the application as it has been killed by LRU or LMK. Thus, phone has to consume more power and time for reopening the applications, rebuild the process and reload the necessary files as prerequisite. Therefore, mobile phones often suffer from high power dissipation, slow starts up of applications and bad user experience. Self-evidently, both LRU and LMK have the same shortage on managing phone memory, and better solutions are looked forward to beat the big challenge. Fortunately, when the ways



concerning user habit are used, this question seems to acquire better solutions (Baik K [1] in 2014, Song W [2] in 2014). In 2016, a solution named Bayesian Network-based Launch Predictor (BNLP) had decreased the restart ratio of LRU by 17.2% [3]. In May of 2018, a solution based on Decision Tree (DT) and Knapsack algorithm [4] can save energy about 38.4%, which increase the power saving by 19.9% than other normal technology such as delayed tolerance and batch processing.

Another problem which makes recent alleged smart phones seemed not to be intelligent enough is the launcher arrangement policy. All the phones of the same model type have same launcher desks and icon indexes after mass production, which cannot change intelligently with user's preference. Even if some of the icons have been rearranged by the user, the order of the desk icons cannot adjusted dynamically to the latest favorite priorities of user. These problems would lead to the difficulties that users have to pay more time and energy to find their recent interested applications. If user's habit of using application can be grasped by the phone and used in the backstage as instructions for reordering launcher icons, this phone would be much more intelligent and popular.

There are still many other mobile applications needing improvement by the user habit learning. For example, the notification system normally lists the new coming messages according to their arriving time stiffly, or some smarter ones would collect and compress the new information of the same application in one line. Nevertheless, it is seriously devoid of a type of solution that always top and highlight the notifications which the user is more interested in. Besides, many useful and lucrative proposal systems and customization application scenes also need the deep research on the user habit oriented to the specific user, so that the phone can offer more characteristic and personalized service based on its user's individual tastes. Therefore, habit learning based on mobile data and more intelligent method is heated since the time of smart phone came (Shye A [5], Do TMT [6], Xu Q [7], Kang JM [8]) and in urgent need now.

In brief, if user habit can be learned by the phone itself naturally and precisely, the problems above would be solved better and the user experience would be satisfied more intelligently. Hence, this topic is researched and developed accordingly in diverse directions. A main stream of learning user habit is to reduce phone power and memory consumption. Methods collecting the resource consumption of application, such as CPU and memory records, network band and device occupation data and using AI model to mine the user habit (Xu Chao [9] in 2015) will be introduced in the following part of the paper. Another major direction is to collect the user network history and text to learn the user need and inclination, so that the sales target can be located more accurate and the customization of the services can be improved (Fang.[10], ZHU.[11]). There are also some other special directions, such as learning the sound or language habit so that user's instructions with unique accent or dialect can be recognized and followed faster and more correctly. For example, Huawei Magic series of phones.

This paper mainly targets on the application improvement. By introducing and comparing relevant algorithms, it recommends a new solution based on "Mode + Period" time division, mining several closely correlative factors of user habit with LT method, constructing and deeply unearthing the DLPA relation between location and application habit. With the location, application information and their inner relationship analysis, the solution calculates conditional probability and predicts the location and application probabilities based on Bayes model. In the end, the special advice on solving LRU and LMK problem based on the learned user habit are offered as reference for improvement of memory management.

2. The Machine Learning Algorithms Targeting to Improve Phone Memory Management based on Application Analysis

As what has been introduced in the first part, phone memory problems caused by LRU and LMK are big challenges to the phone system design, and one important goal of the new solution focused in this paper is also to solve them. In this part, two typical algorithms which have tried to deal with these problems and are also based on machine learning models and application analysis will be introduced below.

2.1 Smart Terminal Resource Cache Algorithm Based on Resource Prediction

It is a resource usage prediction based resource cache algorithm [9]. The applications' resource consumption data is recorded and the resource state and resource bottleneck of the next time interval are predicted by Markov model. The weight of each application is adjusted dynamically and the resource cache problem is converted to multidimensional multiple-choice knapsack problem to minimize the switch time of the application. It is a lightweight heuristic solution algorithm with lower time complicity.

After the resource consumption data of an application is obtained, Markov model is used to calculate its resource consumption according to the algorithm. The status of associated resources used by the application at moment t are recorded by $X_t = \{x_0, x_1, \dots, x_i, \dots, x_t\} (i = 0, 1, \dots, t)$. The algorithm assumes that the status of each resource can be divided by n types (First Assumption), each of which can be described as

$S_n = \{s_1, s_2, \dots, s_j, \dots, s_n\} (j = 1, \dots, n)$. According to Markov assumption:

$$P(x_{t+1} = s_b | x_t = s_a, x_{t-1} = s_c, \dots) = P(x_{t+1} = s_b | x_t = s_a) \quad (1 \leq a, b, c \leq n) \quad (1)$$

It defines the one step transforming probability as $P_{ab}(t) = P(x_t = s_b | x_{t-1} = s_a)$. Based on the assumption that the resource consumption probability has nothing to do with exact time (Second Assumption), the probability $P_{ab}(t)$ can be $P_{ab}(t) = P_{ab}$.

Based assumptions and analysis above, the key steps of this algorithm is as follows:

Step1: Within the window time, the frequencies of each status of all the resource are recorded. That is to say, when $x_{t-1} = s_a$ and $x_t = s_b$, set $C_{ab} = C_{ab} + 1$. Then the resource consumption matrix C can be presented as following:

$$C = \{C_{ab}\} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{pmatrix} \quad (2)$$

Step2: According to the definition of P_{ab} , $P_{ab} = \frac{C_{ab}}{\sum C_{ab}}$, the one step transforming probability matrix can be written to be following form:

$$P = \{P_{ab}\} = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{pmatrix} \quad (3)$$

Step3: According to Markov model:

$$x_t = x_{t-1} \cdot P = x_{t-2} \cdot P^2 = x_{t-3} \cdot P^3 = \dots = x_0 \cdot P^t \quad (4)$$

Thus, only if the P_{ab} can be decided by finite records, the resource consumption of an application at moment t can be predicted and compared with that of other applications.

Step 4: When the resource intension happens, LRU or other resource management system can kill the applications that occupy resource most at moment t if necessary.

[Advantage]: Simulation results show that the precision of the resource usage prediction of the algorithm is superior to others by about 5.4%, and the switch time of the application is reduced by about 45%.

[Shortage]:

2.1.1. Adaptability Limitation. This algorithm is mainly based on analysis to phone resources, such as CPU, memory, storage, decoder, modulator and so on. So it cannot contribute reference to the prediction of other matters, such as user's recent inclination to applications which can help rearrange the launcher icons intelligently.

2.1.2. Target. It mainly collects and analyzes applications' property of using physical devices of phone, not aiming to the habit learning of users. So it doesn't fit the recommendation and customization systems or marketing location which need the prediction of user habit. Its adaptability is relatively narrow and limited.

2.1.3. First Assumption Problem. Its first assumptions requires that the status types of all the resources should be able to presented by n types as $S_n = \{s_1, s_2, \dots, s_j, \dots, s_n\} (j = 1, \dots, n)$. But in reality, different resource have diverse status, it is hard or even impossible to ensure the assumption one all the time.

2.1.4. Second Assumption Problem. An important condition of using Markov model is $P_{ab}(t) = P_{ab}$, It demands that the probability has nothing to do with time. However, the probability that an application need a type of resource is not always definite. For example, for a new bought phone in which there is only 100 contact record loaded from old phone, searching the phone number of a friend needs only to search 100 times at most; while when the contact has 1000 records, searching a friend's number may consume 10 times more CPU and other resource. So, for the same application "contact" and same operation, the resource occupation probability can change along with time actually. Thus, its assumption two is not incontrovertible.

2.1.5. First Insufficiency. In the algorithm, the resource consumption matrix C is recorded within a Window Time. How to decide its length is not mentioned in the algorithm. Evidently, longer window time may produce different result with the shorter one, which would influence the effect and feasibility of the algorithm.

2.1.6. Second Insufficiency. In addition, the resource consumption matrix C records only the frequency of the status, the duration and the quantity of the resource occupation are not taken into account. It is not enough for the target question because these factors significantly affect the left available resource and the resource using performance of the application. Hence, this flaw should be reconsidered by the algorithm as well.

2.2 Application Management Optimization for Smart Terminals Based on User Behavior Analysis

This algorithm is called BNLP, which analyzes users' application using behavior to predict the probability of launching each application in the near future. Then BNLP will manage the background applications according to these probabilities. The main steps of the algorithm is as below.

Step1: Within the specific history window size, collecting the records of last used applications, longitude and latitude, time of day, day of week and the relation between applications.

Step2: Selecting useful information from the records and sequencing them, finding out the common applications.

Step3: Based on BNLP model to score the applications, which indicate the occurrence probability.

Step4: The prediction result will not be used until LRU needs to clear memory and stop some applications. The task killed based on BNLP model goes as following process as Figure 1:

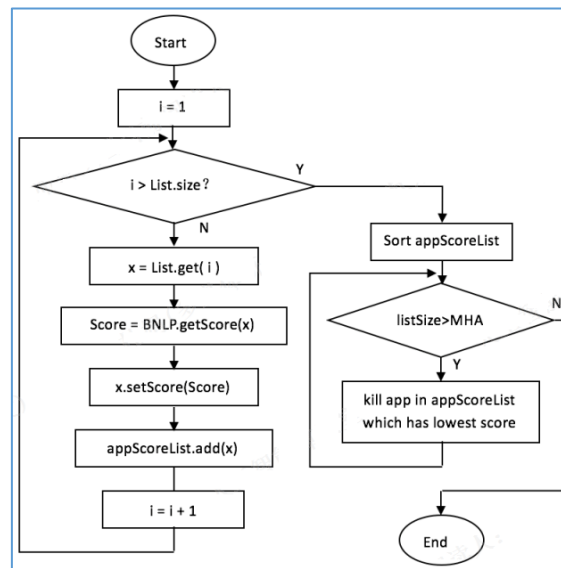


Figure1. The flowchart of task killing mechanism of BNLP algorithm

[*Advantage*] Results show that the model can decrease the restart ratio of LRU by 17.2%, which greatly reduces launch delay, energy consumption and improves user experience.

[*Shortage*]:

2.2.1. Adaptability Limitation. This algorithm is triggered and used only by LRU memory management module. It cannot be widely used for solving other problems by other applications, such as LMK problem. Meanwhile, for the most time when LRU doesn't use the forecast of application, its data collection is useless. However, the GPS orientating always requires great system memory and resource, consumes large quantity of data flow, and even drags down the system response speed. If leaving the GPS on for futile target for long time, the solution would not be able to save energy but increase power consumption because of extra locating waste, then it is likely be prosecuted by users and loses its practicability and value.

2.2.2. Futility. Though it collects data of location, time of day and the day of week, it doesn't dig their inner relation, construct and use the relationship to serve better prediction.

2.2.3. Insufficiency. The algorithm uses the collected data directly without exploring the deeper interrelation of different types of data. This may cause that the conditional probability is not accurate enough for real life judgement and lead to kill the application wrong.

2.2.4. Window Size. Even if the algorithm mentions the effect of window size to the result, it doesn't offer unambiguous definition or suggestion at least to the window size. So, it is not sufficient and rigorous enough.

3. The Machine Learning Solution based on Period and Deep Construction of Mobile Data for Predicting User Habit

To solve the left problem of recent above solutions and offer a more adaptable machine learning solution for obtaining phone user habit, a new algorithm is specially designed. It divides all the time based on the "Mode + Period" principle at first. Within each "period", it uses linked trigger (LT) mechanism and filter policies to collect the data, applies directed location pair application (DLPA) Method to present the underlying relation between the location and applications, combines the location and applications according to special rules. After these multi-constructions and pretreatment, the constructed data are sent to the AI learning layer to learn the location and application using habit. The

learned result is used to predict the probabilities of the location and application in the near future. These probabilities can be used anytime by any type of application which needs the prediction.

3.1 The system architecture

The algorithm has totally six layers, and the system architecture is as Figure 2:

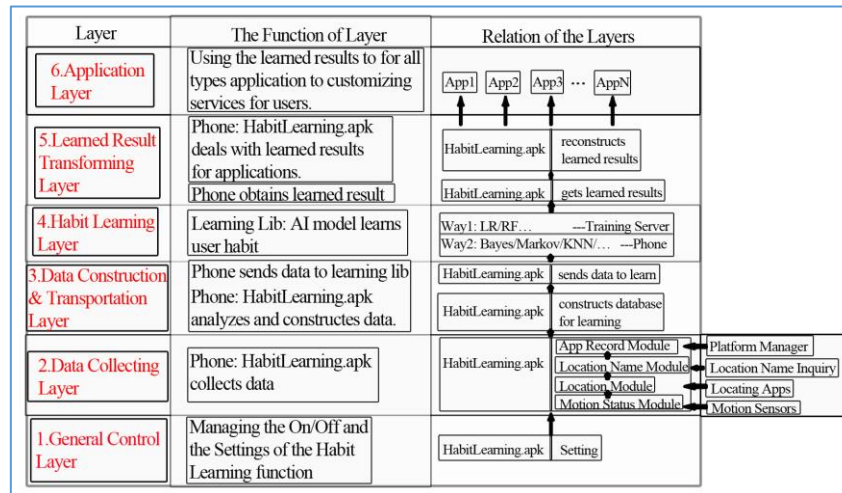


Figure2. The system architecture of the algorithm

3.2 The General Control Layer

This layer has a general manager to switch the whole function of user habit learning on/off so that users can control the energy consumption of their phone conveniently and won't worry about the habit learning function would wastes their data flow or power.

Second import function of this layer is to set default configuration for the whole user habit learning function. As this solution is based on the detailed time division according to "Mode + Period", for which the data collection, data construction, habit learning and learned result usage are all within its own period. For example, the "Workday + '7:00—9:00' Period" is an independent period for the mode "Workday" and the specific period "7:00—9:00" of workday. For this period, the data would be collected and constructed only for this period every day. Monday to Friday are all workdays, so the data in the period "7:00—9:00" of these days would all be collected, saved, constructed, transported to habit learning lib layer and learned by machine learning models. Finally when the learned result comes out, its probabilities of location and application would merely stand for the recent daily routine and application usage habit of user within the period "7:00—9:00" of workday.

One day is a single unit of data circle, and the length of Time window can be set by user according to their satisfaction of the learned result. The default duration of learning data is 30 days, when the duration is longer than that threshold, the system would get the latest 30 days and delete the older data to save storage automatically.

Only if the general switch is on, when the last period ends, the solution will send all the data of the period to "3 data construction layer" to construct. At the same time, the system would start the data collection of the next period by itself. When one day comes to the end, all the data of the periods of the day would have already been sent to the construction layer and learned by the habit learning lib.

The default setting of the modes is workday and weekend, which have seven periods and four periods respectively. Workday is from Monday to Friday, and Weekend includes Saturday and Sunday. Seven periods of workday are 0:00—7:00, 7:00—9:00, 9:00—12:00, 12:00—13:30, 13:30—18:00, 18:00—21:00, 21:00—24:00. Four periods of weekend are 0:00—8:00, 8:00—12:00, 12:00—18:00, 18:00—24:00. These modes and periods can all be changed by user on the interface of Habit Learning application developed specially for the whole function. This application contains the general switch inside as well, while the "3 Data construction and transportation layer", "4 Habit learning Layer" and

“5 Learned result transforming Layer” are all compressed into habit learning lib and embedded in the habit learning application. The function separation is as following Figure 3:

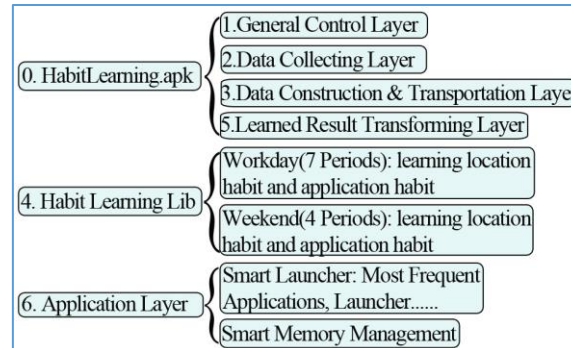


Figure 3. The function separation of the algorithm

3.3 The Data Collection Layer

In this layer, the user motion status, location longitude and latitude, location name and application history are obtained by LT mechanism to save energy and reduce unnecessary operation. During these collection, there are some policies to filter the useless data and save time and power, too.

The sequence of linked trigger (LT) of data collection is as following:

Step1: The motion sensor monitoring the user motion status, the location module will not start positioning until the motion status changes and notifies the location module.

Step2: Once the location module starts to locate, the filter start to work. Policy one of the filter is for ignore the location data which are too close, if the adjacent location data are mined from two locations whose distance is less than a minimum threshold, new record would not be generated. But in order to make full use of this location, the accuracy of the new data is checked: if its accuracy is higher than the older location data, use its longitude, latitude and accuracy to replace the old one, or else give up it.

Step3: When the location data is decided, trigger the location name module. Use the longitude and latitude and special map to get the location name. Here the second policy of filter works to delete the useless location name. If the former location and the new location have the same name, that shows two possibilities: one is the user indeed stay in one place within the period, then if the duration of the two locations record are more than a duration threshold, accept it as correct record. If the duration is too short, it will be given up as illegal location, because that means user stays in the same place for a too short period of time, there is no need to generate a new record. Then the records of location can be presented as following location matrix X:

$$X = (X_0, X_1, \dots, X_i, \dots, X_M) = \begin{pmatrix} \text{ModePeriod} & \text{Longitude1} & \text{Latitude1} & \text{Location Name1} & \dots & \text{Accuracy1} \\ \text{ModePeriod} & \text{Longitude2} & \text{Latitude2} & \text{Location Name2} & \dots & \text{Accuracy2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \text{ModePeriod} & \text{LongitudeM} & \text{LatitudeM} & \text{Location Name M} & \dots & \text{Accuracy M} \end{pmatrix} \quad (5)$$

Step4: Getting the application history at the end of the period and send it together with the location data to the “3 Data Construction & Transportation Layer”. Following shows the data structure of the application:

$$Y = (Y_0, Y_1, \dots, Y_j, \dots, Y_N) = \begin{pmatrix} \text{ModePeriod} & \text{App Name1} & \text{Longitude1} & \text{Latitude1} & \text{Location Name1} & \dots & \text{Accuracy1} \\ \text{ModePeriod} & \text{App Name2} & \text{Longitude2} & \text{Latitude2} & \text{Location Name2} & \dots & \text{Accuracy2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \text{ModePeriod} & \text{App NameM} & \text{LongitudeM} & \text{LatitudeM} & \text{Location Name M} & \dots & \text{Accuracy M} \end{pmatrix} \quad (6)$$

By the LT mechanism, the model collected the data closely associated with the habit learning. At the same time, it avoids getting the futile data and wasting the system energy and memory by economic and scientific policies mentioned above in the LT mechanism.

3.4 The Date Construction and Transportation Layer

The function of this layer is mainly to receive the data collected in the former layer, construct the data in special form and combine the data according to the requirement of the AI model in habit learning lib. There are also several policies in this layer to combine the similar locations and application for each period. The final data sent to the habit learning lib for training and predicting is as Figure 4 as follows:

Mode	Time	Day	Weekday	Location(Apps, Total Duration, Frequency)			
				Home→Metro Station	Metro Station→Bus Station	Bus Station→HisenseR&D	
Work day	7:00 — 9:00	1	Monday	News 20m, 2	CheLaile 10m, 2	Words 15m, 2	
				Weather 5m, 1	Music 10m, 1	Dictionary 5m, 3	
		2	Tuesday	News 15m, 2	CheLaile 10m, 2	Wechat 10m, 2	
				Music 20m, 1	Call 5m, 1	Weather 5m, 1	
		3	Wednesday	News 15m, 1	CheLaile 5m, 1	Wechat 10m, 2	
				Call 5m, 1	Wechat 10m, 2	Music 20m, 1	
		4	Thursday	Weather 5m, 1	Music 10m, 1	Wechat 10m, 2	
				Music 10m, 2	Wechat 10m, 2	Alarm 3m, 1	
		5	Friday	Music 15m, 1	CheLaile 5m, 1	Wechat 15m, 3	
				Call 5m, 1	Toutiao 10m, 2	Video 10m, 2	
						Call 2m, 1	

Figure 4. An example of the data constructed in the Data Construction and Transportation Layer

3.5 The Habit Learning Layer

This is the core of the whole model where the AI model of Bayes is used to learn the location and application habits for each period. The usage of Bayes model is based on some basic assumptions and analysis.

3.5.1 The Basic Assumptions. The target of this model is the user habit, specially the user's location habit and application habit. We learn these habits based on following assumptions:

Assumption1: In the different "Day Mode", user normally has disparate daily schedule. Accordingly, the routine of the day would be different from that the other different "Day Mode", and their location chains would be likely different;

Assumption2: In the same "Day Mode", user generally shares similar schedule or life mode. Under such schedule with high similarity, user would probably follow the similar routine and location chain. So the directed location pair would be with high likeness.

Based on the two assumptions above, a hidden rule of user daily habit can be disclosed as following: Day Mode decides the daily schedule, and daily schedule decides the routine and location chain, accordingly, location chain decides the directed location pair. The interrelation of these factors can be shown as:

Day Mode → Daily Schedule → Daily Routine

Assumption3: Even in the same day mode, people have different habit in different period of time. For example, in the time period of "7:00am---9:00am" of workday, people usually are on the way to company; while in the time period of "9:00am---12:00am" of workday, though it is still in the same "Day Mode", people usually are working in the office. Thus, the time period decides the specific location chain of the user, and accordingly indirectly decides the directed location pair.

Summarizing the three assumptions above, following implicit relation can be revealed:

"Mode + Period" → Daily Schedule → Daily Location Chain → Daily Directed Location Pairs

Note: Location pair should be directed, because location A to B stands for different direction and aim or task for the location pair. For instance, people go from metro station to bus station usually for going to work in the morning, but when the phone submits his location is from the bus station, it is

likely for different aim or task, such as going back home for urgent matter or go to other places. Therefore, location $A \rightarrow B$ indicates different aim or meaning for the user habit from that of $A \leftarrow B$, and this model differentiates this key point with the creative name “**Directed Location Pair (DLP)**”.

Assumption4: When the “Mode + Period” is decided, user always goes from $A \rightarrow B$ for similar target. Accordingly, user would probably use similar applications during the time period and between the location pair. The applications on this directed location pair are called “Directed Location Pair Application (DLPA)”.

This assumption is easy to understand as well. For example, in the 7:00---9:00 of workday, a user who usually on the way going to company would likely go from the metro station to the bus station for taking bus. With this target of the directed location pair “metro station \rightarrow the bus station (workday, 7:00---9:00)”, the user always checks the “Bus Inquiry” application for the bus information because he wants to take some bus he takes every day. So, it is self-evident that the occurrence of the application “Bus Inquiry” in this “Mode + Period” has close thing to do with the directed location pair “metro station \rightarrow the bus station (workday, 7:00---9:00)”. This is essential to the prediction of the location pair: because once the application “Bus Inquiry” appears in this “Mode + Period”, it indicates great probability that the user is on the way from metro station to the bus station. In another word, the appearance of the application offers convincing evidence or great information of his location and direction. Therefore, the DLPA collected by the habit learning model contain important information for predict user’s location habit; Likewise, if the directed location pair is detected by the habit learning model, it shows great possibility that the user would probably open the “Bus Inquiry” for the location pair of “metro station \rightarrow the bus station (workday, 7:00---9:00)”. That’s to say, the directed location pair can be used for predicting the applications which are likely to be opened in the near future.

All the information and implication contained by the important newly innovated concept DLPA in this paper is significant finding of this habit learning model, and it can be used for the user daily location and application prediction. What’s more, this kind of implicit correlation between location and application of user habit are mutual conditions for the occurrence of the two factors, which involves a kind of conditional possibility relationship and can be predict with conditional probability.

3.5.2. The Introduction of Bayesian Network Model Directed graphical model is also called Bayesian network. All sides are directed, and the directions of the arrows indicate the probabilities of the stochastic variables are decided by other stochastic variables. The probability can be shown as following:

$$P(x) = \prod P(x_i | \text{Pag}(x_i)) \quad (7)$$

$\text{Pag}(x_i)$ stands for all the father nodes of x_i . Following is an example of Bayes model as Figure 5 below:

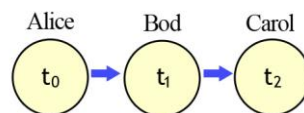


Figure 5. An example for showing Bayesian Network

In the figure above, t_1 depends on t_0 ; t_2 depends on t_1 , and indirectly depends on t_0 . Their relationship can be expressed as following:

$$P(t_0, t_1, t_2) = P(t_0)P(t_1|t_0)P(t_2|t_1) \quad (8)$$

Only if each variable of the graph has small amount of father nodes, the probability would be able to be calculated by several parameters. Bayesian Principle is as follows:

$$P(c|x) = \frac{P(x|c)P(c)}{\sum P(x)} \quad (9)$$

Bayesian network model is a model for conditional possibility. This is why we choose Bayes as model base.

4. Conclusion

At first, this paper introduces two algorithms for improving the phone memory management: the first is the prediction of all types of device and resource consumption based on Markov model; the second is based on BNLP. Both of them have their shortcomings, such as the limited adaptability of the algorithm, the insufficient application of the collected data, the ignorance of the consideration of the power and energy consumption of the model, and the imprecision of the prediction. By contrast, the innovative model offered later in the paper, which specially learns the user habit with machine learning algorithm based on deeply constructed phone data, can solve all these left problems perfectly. It explores deeply with the implicit interrelation between the location and application and uncovers it with DLPA, analyzing the conditional possibility solution for the problem and finding the suitable model Bayes for it. Besides, the solution of this paper designs brand new system architecture with six layers, creating a LT data collection method for saving energy and getting most useful relative data with mobile phone. This thorough and vigorous consideration equips the algorithm with comprehensive reference for the problem on the user habit learning direction.

Reference:

- [1] Baik K, Huh J. Balanced memory management for smartphones based on adaptive background app management. The 18th IEEE International Symposium on Consumer Electronics. IEEE. 2014. 1-2.
- [2] Song W, Kim Y, Kim H, et al. Personalized optimization for Android smartphones. ACM Trans. on Embedded Computing. Software Technique. Algorithm Systems (TECS), 2014, 13(2s): 60.
- [3] HUANG Wen-Qian, LI Mei, YU Jia-Geng, (2016) "Application Management Optimization for Smart Terminals Based on User Behavior Analysis." Computer Systematic Application. 25.10: 201--206.
- [4] ZHU Zhengwei, LIU Chen, HUANG Xiaozhu, et al. Smartphone Energy Consumption Optimization Method Based on User Behavior [J]. Computer Engineering, 2018, 44(5): 286-290.
- [5] Shye A, Scholbrock B, Memik G, et al. Characterizing and modeling user activity on smartphones: Summary. ACM SIGMETRICS Performance Evaluation Review. ACM, 2010, 38(1): 375–376.
- [6] Do TMT, Blom J, Gatica-Perez D. Smartphone usage in the wild: A large-scale analysis of applications and context. Proc. of the 13th International Conference on Multimodal Interfaces. ACM. 2011. 353–360.
- [7] Xu Q, Erman J, Gerber A, et al. Identifying diverse usage behaviors of smartphone apps. Proc. of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference. ACM. 2011. 329–344.
- [8] Kang JM, Seo S, Hong JWK. Usage pattern analysis of smartphones. Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific. IEEE. 2011.1-8.
- [9] Xu Chao,Zeng Xuewen,Guo Zhichuan. Smart Terminal Resource Cache Algorithm Based on Resource Prediction[J]. Computer Engineering,2015,41(3):59-63.
- [10] Fang Z X, Yu C, Zhang T, et al. 2017. A mixed Markov method to predict the surfing time period of mobile phone users . Journal of Geo-information Science, 19(8):1019-1025. SP.J.1047.2017.01019
- [11] ZHU Zhengwei, LIU Chen, HUANG Xiaozhu, et al. Research on Prediction of User Behavior When Using Network on Mobile Phones [J]. Computer Measurement and Control, 2017, 25(11): 212-218.