Prophet: What App You Wish to Use Next

Xun Zou
Zhejiang University
38 Zheda Road
Hangzhou, 310027 PRC
zouxun@zju.edu.cn

Gang Pan*
Zhejiang University
38 Zheda Road
Hangzhou, 310027 PRC
gpan@zju.edu.cn

Wangsheng Zhang
Zhejiang University
38 Zheda Road
Hangzhou, 310027 PRC
zws10@zju.edu.cn

Shijian Li
Zhejiang University
38 Zheda Road
Hangzhou, 310027 PRC
shijianli@zju.edu.cn

*Corresponding author.

UbiComp '13, Sep 08-12 2013, Zurich, Switzerland.
ACM 978-1-4503-2215-7/13/09.
http://dx.doi.org/10.1145/2494091.2494146

Abstract
A variety of applications (app) installed on smart phones do greatly enrich our lives, but make it more difficult to organize our screens and folders. Predicting apps that will be in use next can benefit users a lot. In this poster, we propose some light-weighted Bayesian methods to predict the next app based on the app usage history. The evaluation on Mobile Data Challenge (MDC) dataset gives very encouraging results. In addition, we suggest a natural way to integrate the app prediction features to the user interface. Users would find it convenient to access the predicted apps with simple touches.

Author Keywords
Smart Phone; App Prediction; Context; User Interface.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Smart phone applications have boomed in recent years. By May 16, 2013, the app downloads from App Store had exceeded 50 billion. As a comparison, 48 billion apps have been downloaded on Android, according to the statements on Google I/O 2013. With so many apps downloaded and installed on smart phones nowadays,

users have to organize their apps, folders or menus more frequently. In addition, more time is spent locating and selecting the apps they want to use. So, it is becoming quite an important issue for user interfaces that how to help users find what they want as quickly as possible.

Several methods have been proposed in previous work on app prediction. Huang et al. [2] used contextual information to build models on a MDC dataset [3] composed of 38 users, and found the latest used app the strongest predictor. Yan et al. [7] used contexts such as user location and temporal access patterns to help prediction. Based on naive Bayes classifiers, Shin et al. [5] developed a novel dynamic home screen application (*dynamic home*) on Android. It indeed made it slightly faster for users to locate apps. However, some users were unaccustomed to its letting apps appear and disappear unexpectedly.

In this poster, three Bayesian models are proposed which perform slightly better than existing methods. Based on the feature of IOS, we also suggest a natural way to integrate the app prediction features. This allows users to access predicted apps easily with gestures and can avoid the problem caused by *dynamic home* at the same time.

App Prediction Model

According to the previous research [2, 5], location and other context information made little or even negative effects to prediction accuracy when the latest used app and time are used. Thus we rely only on app usage history and time context. We implement the Most Frequently Used Model (MFU) and the Latest Used Model (LU) as bench marks. As more complex models require much more computing resource which is infeasible on mobile devices, here we focus on three Bayes methods to do prediction.

Consider a user whose app usage history is $a_1a_2...a_{n-1}a_n$, and think of the user's app as a random variable X. Now we get $a_1a_2...a_{n-1} = X_1X_2...X_{n-1}$, and want to predict a_n. Let A be the set of apps, LU calculates a score for each X in A:

$$Score(X) = p(a_n=X \mid a_{n-1}=X_{n-1})$$

The apps with highest scores are selected as a_n.

- *the Latest Used Apps (LU-2)*

Here we take both X_{n-1} and X_{n-2} into consideration:

$$Score(X) = p(a_n=X \mid a_{n-1}=X_{n-1}, a_{n-2}=X_{n-2})$$

If the highest scores are 0, then calculate scores as LU. This is introduced as *"fallback"* Markov model in [6].

- *the Latest Used App with Time (LUT)*

If the time interval between a_n and a_{n-1} is too large, then X_n will not have much to do with X_{n-1}. So we can predict as LU if the interval is within a certain threshold, and use MFU otherwise.

- *Bayesian Network (BN)*

Since X_{n-2} is found useful in LU-2, we try a linear combination of X_{n-1} and X_{n-2}:

$$Score(X) = \mu p(a_n=X \mid a_{n-1}=X_{n-1}) + (1-\mu)p(a_n=X \mid a_{n-2}=X_{n-2})$$

At first, we make μ increased by 0.1 each time from 0 to 1 and get a rough value for μ. Then we search its neighborhood for better values.

Evaluation

The dataset we use for evaluation comes from *Set A* in MDC [3], which contains 80 users' app records. First of all, we eliminate apps that are idle or system process.[1] After the preprocessing, we get an average of 59 active apps and 5589 records for each user. We apply 10-fold cross validation, and use the overall hit rate for evaluation:

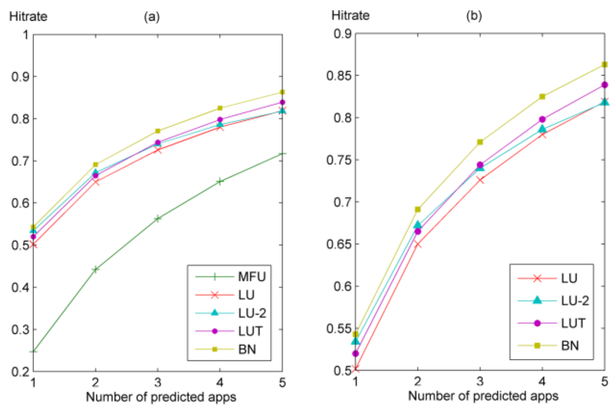

Figure 1. The left figure (a) displays the results of all the five predictors. MFU is eliminated in the right figure (b) to make the details clearer.

From the figures, we can observe that BN achieves the highest accuracy within 5 predicted apps. Both LU-2 and LUT slightly outperform LU. However, BN and LUT should spend time learning the parameters while LU-2 is quite light-weighted and easy to implement. But the hit rate of LU-2 declines compared to other methods when the number of predicted apps increases.

1 Apps with UIDs: 101fd64c, 100056cf [2], 101f4cd2.

Analysis

- *LU-2 and LU-k*

Since LU-2 has advantage in time over the other 2 methods and performs quite well, will LU-k do better? However, accuracy does not increase with larger k. In general, LU-k predicts with the k apps before X_n. We refer to the k apps as a k-pattern, and each pattern is a point in A^k. With k growing bigger, the dimension of A^k increases while the total number of patterns remains the same. By projecting these patterns to space of higher order, the number of each pattern decreases. That's to say, the current pattern $X_{n-k}...X_{n-2}X_{n-1}$ can hardly be found in the training set. And then our LU-k predictor which bases on probability becomes ineffective.

- *the time interval in LUT*

In previous experiments, we increase the time interval from several seconds to one day. The optimal time intervals vary for different users, and they are either small or very large. Among the 80 users, the intervals for 30 users are relatively small. They range from several minutes to less than two hours. This seems to fit most people's app usage pattern, as the app we use now often has nothing to do with the one we used hours ago. For the other 50 users, their optimal intervals are as long as one day, which indicates that LU always has better performance than MFU for these users.

- *the number of items in BN*

Similar to LU-k, what if $p(a_n=X \mid a_{n-k}=X_{n-k})$ is added to the score calculation? With more than two items in the expression, we can learn the parameters through least square techniques. However, the accuracy also falls

Figure 2. Predicted apps bar is shown on the top of the screen.

Figure 3. Predicted apps are placed above the object folder.

with k larger than 2. One possible reason is that, the more items added, the more difficult the parameters' learning is.

User Interface Design

Touchscreen and motion sensors greatly enhance smartphone's interaction capability [1, 4]. Here we mainly focus on the IOS. As we all know, IOS supports gesture operations so that users can access the latest used apps conveniently. Say we can use four or five fingers to swipe up to reveal multitasking bar and left or right to switch between the latest used apps. This seems to be a natural feature to combine with app prediction.

When swiping up or down, we can put the predicted apps on the top of the screen (Figure 2). Similar improvement can be done on the folders. When a folder is touched, the apps in it will be displayed below it. Among these, we can put the apps that are most likely to be used above the folder (Figure 3). This design gets rid of the problem caused by *dynamic home*, as we provide both predicted apps and the original layout. Besides, it is also quite cool to reveal the predicted apps with gestures.

Conclusion

In this poster, we proposed three Bayes models on app prediction, and found they slightly improve the hit rate on the MDC dataset, compared to the Latest Used Model. We also suggest a novel user interface that integrates app prediction features on the IOS. Our next step is to implement this design and collect users' feedback about it. With corresponding data, we are able to achieve higher prediction hit rate and make better user experience.

Acknowledgements

This work is supported in part by the National Key Technology R&D Program (2012BAH25B01, 2012BAH94F01). The authors would like to thank Nokia, especially the MDC research team, for making such a rich and precious dataset available to the researchers.

References

[1] Du, Y., Ren, H., Pan, G., and Li, S. Tilt & Touch: Mobile Phone for 3D Interaction. In *Proc. UbiComp 2011,* ACM Press (2011), 485-485.

[2] Huang, K., Zhang, C., Ma, X., and Chen, G. Predicting mobile application usage using contextual information. In *Proc. UbiComp 2012,* ACM Press (2012), 1059-1065.

[3] Laurila, J. K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T. M. T., ... and Miettinen, M. The mobile data challenge: Big data for mobile computing research. In *Proc. Pervasive 2012,* 1-8.

[4] Pan, G., Ren, H., Hua, W., Zheng, Q., and Li, S. EasyPointer: What You Point at is What You Get. In *Proc. SIGCHI 2011,* ACM Press (2011), 499-502.

[5] Shin, C., Hong, J. H., and Dey, A. K. Understanding and prediction of mobile application usage for smart phones. In *Proc. UbiComp 2012,* ACM Press (2012), 173-182.

[6] Song, L., Kotz, D., Jain, R., and He, X. Evaluating location predictors with extensive Wi-Fi mobility data. In *INFOCOM 2004*, IEEE, Vol. 2, 1414-1424

[7] Yan, T., Chu, D., Ganesan, D., Kansal, A., and Liu, J. Fast App Launching for Mobile Devices Using Predictive User Context. In *Proc. MobiSys 2012,* ACM Press (2012), 113-126.