Influencer Innovator Smartphone

Collaboration Projects

Using Smartphone Usage Histories to Identify Influencers and Predict Application Adoption

This article presents a research trial conducted to predict future usage of smartphone applications by analyzing the influence among individuals through their usage histories. In this research, we modeled the level of influence between people based on the temporal sequence in which users download and execute applications, and have hypothesized that latent groups can form in these influence relationships. By collecting usage histories from approximately 160 university students, we have verified this hypothesis and shown that it enables highly accurate predictions. This study was conducted jointly with Cybermedia Center, Osaka University, in a joint research division set up in the center.

Service & Solution Development Department
Masaji Katagiri
Minoru Etoh†

1. Introduction

With the rapid popularization of smartphones, the number of applications available for the Android™*1 operating system surpassed 400,000 in 2011, and these applications are set to become even more diversified and numerous. Thus, application recommendations are of crucial importance in how users select applications, and are therefore a key to furthering growth in the smartphone industry.

Accordingly in recent times, much attention has been focused on marketing methods that are designed to popularize products and services efficiently by seeking out and focusing on users who have strong influence on their peers (influencers). This is because identifying strong influencers that can affect people around them is said to be an effective way to predict trends, expand business and increase revenues etc (see **Figure 1**).

In this article, we describe how we have attempted to achieve more effective smartphone application recommendations by applying the influencer concept. Specifically, we have hypothesized a model to express the level of influence between individuals based on the temporal order in which users

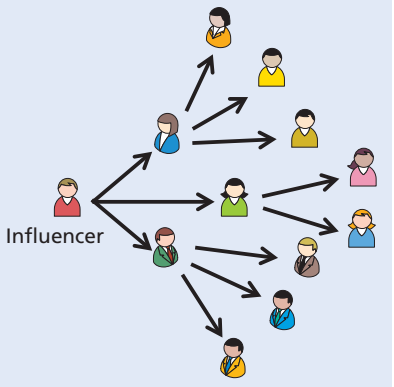


Figure 1 The influencer effect

† Currently in the R&D Strategy Department

*1 **Android™**: An open source platform targeted mainly at mobile terminals or promoted by Google Inc., in the United States. Android™ is a trademark or registered trademark of Google Inc., in the United States.

download and execute applications, to identify the influential relationships between users from their smartphone application usage histories. By using these influential relationships to predict application use, we have achieved better prediction capabilities.

Jointly with Osaka University, the authors conducted usage monitoring experiments carried out on approximately 160 students over a six-month period. Using the usage histories collected, we have empirically evaluated the capabilities of the model we propose, and verified our hypothesis. The results indicate that latent groups exist in the influential relationships among users, and that using these groups to predict application downloads offers superior predictive power compared to conventional prediction methods. Please refer to [1] for more details.

2. Modeling Influential Relationships

2.1 Influential Relationships Between Individuals

Presumably in situations where applications are first used, the application might be introduced to the user by a friend or acquaintance. Obviously however, there are many other ways that users might be persuaded to begin using applications, for example, due to information obtained through the mass media. For this reason, a model of influential relationships which accounts for only direct exchange of information between friends and acquaintances is incomplete. Thus, we have considered the implicit influence that is passed on to people who have no direct relationship with the influencer, and have developed a model that includes people who are involved in relationships of both direct and implicit influence.

Let's consider user pairs in which one of the users will often use a new application first, after which the other user will follow. These are examples of strong implicit influence between two people, no matter whether there exists any direct information exchange between them. The effectiveness of predicting purchase by focusing on this type of sequential relationship between people has been discussed by Kawamae et al [2].

A typical recommendation system offers a user a list of applications that have been predicted to be potentially downloaded by the user in the near future. Today's most common methods for recommendation are Collaborative Filtering (CF)[*2] methods, for which a range of extensions have been researched [3].

Figure 2 shows an example of the temporal relationships in application downloads. Fig. 2 includes three users and five applications, and the horizontal axis indicates the time. The numbers in circles indicate the different applications, and the time that the users downloaded them is shown by their position on the horizontal axis. Here, let's consider which application will be downloaded by user 1. In CF, temporal information is not considered, so there is no difference between users 2 and 3 in terms of similarity with user 1. Therefore applications 4 and 5 will be recommended equally as applications that user 1 might subsequently download. However, if focus placed on the temporal order of downloads, it is clear that user 2 is leading user 1, and is therefore an influencer, while user 3 follows user 1. For this reason, it is obvious that application 5 is more likely to be down-

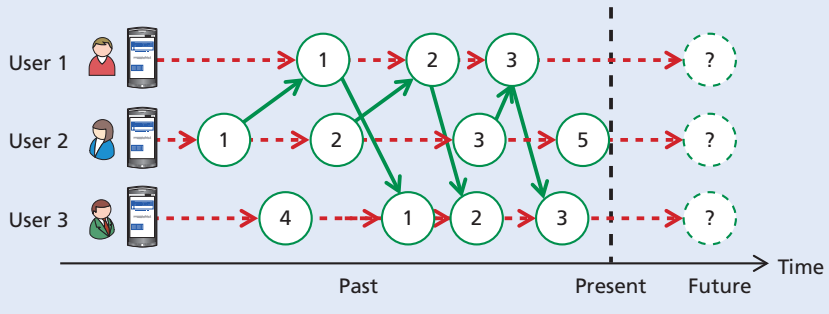


Figure 2 Application usage sequences and predictions

*2 **Collaborative filtering**: Firstly, preferential data such as purchasing history is collected for a number of users, then, using information about the target user and other users with similar preferences, predictions and recommendations are made for those users.

loaded by user 1 than application 4.

As this example clearly indicates, when there are leader/follower relationships in the actions of users, it is quite effective to use the influences between individuals for recommendations. Even intuitively, users who have high sense of information are likely to become early adopters of an application, and naturally general users will follow, a fact which holds promise for marketing effectiveness.

2.2 Model Formulation

Here, we have formulated a probability model based on Bernoulli trials[*3] to describe the occurrence of application download cascades. There are many varieties of applications ranging from the very popular with large numbers of users to niche applications with extremely small numbers of users. In order to grasp influential relationships that more clearly reflect individual preferences, we have defined the influence that user u has over user v as $P_r(u \rightarrow v)$, with entropy weighting, as described in Equation (1).

$$P_r(u \rightarrow v) = \frac{\sum_{a \in A_{u \rightarrow v}} (-\log(u_a / u_{glob}))}{\sum_{a \in A_u} (-\log(u_a / u_{glob}))} \quad (1)$$

Here, "A_u" is the set of applications downloaded by u, $A_{u \rightarrow v}$ is the set of applications downloaded by u ahead of v. u_a and u_{glob} show the number of users of application a and the total number of users respectively. Let the influence matrix R be the matrix with the size [number of users x number of users], where row u and column v element represents the influence between users u and v; $P_r(u \rightarrow v)$.

According to the concept of the independent cascade model[*4] [4], using the influence matrix R obtained, the joint probability that user u will download application d can be described by Equation (2).

$$P_{inf}(d|u) = \left[1 - \prod_{u'} \{1 - P_r(u' \rightarrow u)\} \right] \quad (2)$$

Here, u' is a user who downloaded d before u.

Because $P_{inf}(d|u)$ is the predicted probability that user u will download application d, effective recommendations can be made using a list of d applications that have a large $P_{inf}(d|u)$ value.

3. Latent Group Structures

When calculating predictions from history according to the model described in section 2, the number of applications commonly downloaded by both users must be above a certain number, in order to derive influence relationships reliably for all user pairs. However, in many cases there may not be enough numbers of common applications in all user pairs. Moreover there is not much history available for new users. To deal with this situation, Kawamae et al [2] described a method which assumes a Markov process[*5] and applies ergodicity[*6]. However, this method attempts to include relationships as multi-staged cascades, and cannot compensate for insufficient observations. In our study, by hypothesizing structures of latent groups in mutually influential relationships, the apparent influential relationships can be reliably inferred, which offers better predictive power and thus better recommendation capabilities. As far as we know, no such similar approaches have been reported.

Generally in the field of marketing, user segmentations are commonly used to grasp the preferences and behaviors of people. Accordingly, we hypothesized that such user segments also exist as latent groups for application downloading, which we have verified. The use of latent group structures enabled compensation for the behaviors, since similar behaviors are expected to occur in each group. This method offers reliable predictive capabilities even when not much data for individual users is available. The following section describes the usage history data we used for verification.

3.1 Usage History Data

In the joint research with Osaka University, we lent smart phones (Xperia™[*7]) loaded with specifically designed software to record usage histories to approximately 160 university students for a period of six months. We instructed the students to use the phones freely and collected their usage history.

*3 **Bernoulli trial**: The most basic probability model. A representative example is flipping a coin, in which there is a fixed probability that the coin will land heads up, and then observing which side of the coin faces up when it lands.

*4 **Independent cascade model**: A common probability model used to describe the propagation/diffusion phenomena, such as the spread of information. This model expresses the occurrence probability of propagation chains, where they occur independently and stochastically.

*5 **Markov process**: A process, where the next state can be probabilistically determined from only the current state (past states have no effect) by a transition graph, assuming a limited number of possible defined states.

Participants in the experiment were recruited on the university campus. We explained the purpose of the experiment, and obtained permission to collect the data which include personal information, to handle the data with privacy protection, and use it for research purposes.

Usage histories were anonymized in the handsets, and collected on a server via the 3G network. Each usage history record consists of three items - time, an anonymous user ID and the package name of the executed Android application. After collecting usage histories, we extracted the records of the first execution for each application by each user. **Figure 3** shows the trend in cumulative counts of the first-time execution records from the beginning of the experiment.

We extracted an 89-day training set[*8] (February to April 2011), and a 31-day test set[*9] (May 2011) for evaluating the model, by considering the pattern in Fig. 3. We also eliminated applications which had less than three users from the data sets, as these cannot be part of valid chains. These processes resulted in 3,383 records in the training set (155 users, 291 applications), and 249 records in the test set (98 users, 116 applications).

3.2 Verifying Latent Groups Structures

To extract latent groups, we applied matrix factorization[*10] on the influence matrix R to make low-rank approximations[*11]. Because the elements of influence matrix R represent the occurrence probability of application download cascades, the values of these elements must be non-negative. Therefore, we constrained the values of the elements to be non-negative using Non-negative Matrix Factorization (NMF)[*12] [5]. For NMF regularization terms, we tried commonly suggested terms, and used the ones that yielded the best results.

To verify our hypothesis, we varied the number of groups and observed the accuracy of application download prediction. **Figure 4** shows the results. To measure the accuracy of predictions, we employed perplexity[*13]. The lower the

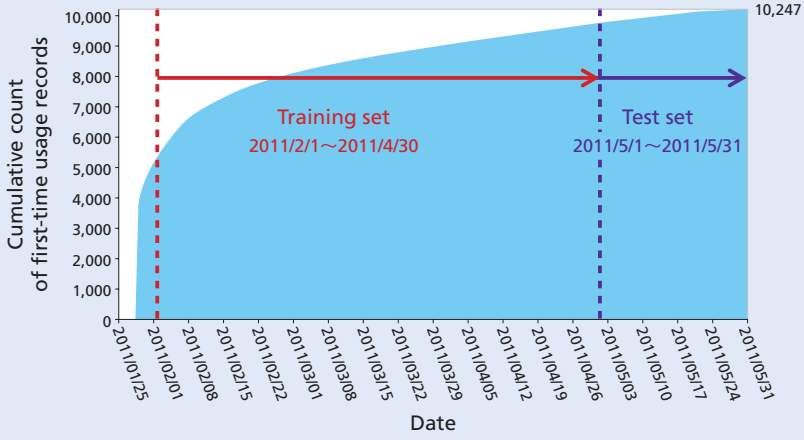


Figure 3 Volumes of usage history records

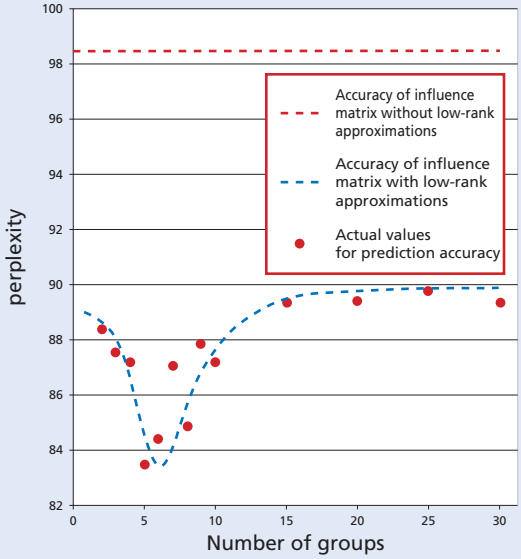


Figure 4 Results of low-rank approximations

*6 **Ergodicity**: In Markov processes, ergodicity is a characteristic of systems to converge to a certain probabilistic distribution among states after many transitions regardless of the initial state.

*7 **Xperia™**: A trademark and registered trademark of Sony Mobile Communications AB

*8 **Training set**: A training set is data used to acquire characteristics, which is prepared separately from test set data. Prediction methods are trained (internal parameters are estimated) based on the training set first, and then tested using the new data (test set data).

*9 **Test set**: A test set is separate data from the training set, which is used for evaluating predictive power by cross validation.

value of perplexity is, the more accurate the prediction. Generally, in absence of latent group structures, the greater the number of groups yields better expressiveness and thus better the accuracy of predictions. In contrast however, we can find a peak in predictive accuracy around six groups, as shown in Fig. 4. This indicates the existence of latent groups in influential relationships.

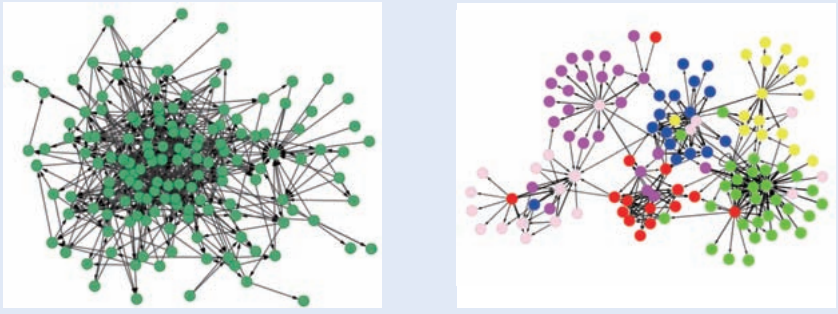
Next, we attempted to visualize the relationships indicated by the influence matrix R, as shown in **Figure 5** (a) for results without low-rank approximations, and Fig. 5 (b) for results with low rank approximations. Each node represents a user, while the edges between the nodes represent influence above a certain threshold value. Fig. 5 (a) and (b) were drawn using a force-directed layout[*14] algorithm - a common graph visualization method - to visualize relationships with binarized influence occurrence probability at the threshold. The node colors in Fig. 5 (b) indicate the latent groups to which users belong. Influencers are represented by nodes that have edges towards many other nodes on the graph. These graphs confirm that the application of low-rank approximations makes it easier to extract group structures that are otherwise not easily visible. Moreover in Fig. 5 (b), we can see the mixtures of influential relationships within and between latent groups.

4. Predictive Power Comparison

We compared the prediction accuracy using the influence matrix, against common existing methods (popularity ranking method, user-based CF — a common CF method). We assessed prediction accuracy using perplexity, as in the previous section. **Figure 6** shows the results. The influence matrix without low-rank approximation applied results in relatively poor prediction accuracy. In contrast, applying low-rank approximations to the influence relationships achieves better results than existing methods. These results confirm the effectiveness of our proposed method.

5. Conclusion

This article has described a method to predict users' future application downloads by using influential relationships between individuals. We have empirically confirmed the existence of latent group structures using usage histories obtained by experiments conduct-



(a) Without low-rank approximations (b) With low-rank approximations

Figure 5 Visualized influence structures

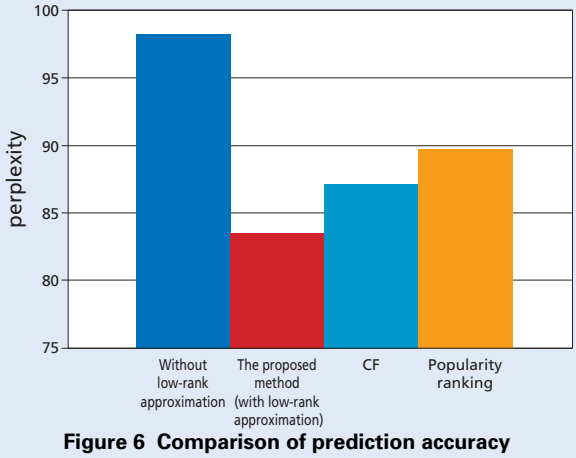


Figure 6 Comparison of prediction accuracy

*10 **Matrix factorization**: Finding the factors of an (m × n) matrix by approximating the matrix by the product of two matrices (m × k) and (k × n), (normally k<<m, n).

*11 **Low-rank approximation**: Approximating any matrix with a lower rank matrix. The product of the two matrices obtained by matrix factorization with dimension K becomes a rank K approximate matrix.

*12 **NMF**: One method of matrix factorization. Matrix factorization in which all elements of the given matrix and the matrices obtained from factorization are constrained to have non-negative values.

ed on 160 university students over a six-month period. The latent group structures enable better predictions about application usage than existing methods.

As for further studies, we intend to develop efficient processing methods to deal with large-scale data, and proceed with verification of applicability and effectiveness in the field. In terms of marketing policies, we left applying influencers obtained from the influence relationships in the list of remaining issues that need to be addressed.

REFERENCES

[1] M. Katagiri and M. Etoh: "Social Influence Modeling on Smartphone Usage," Proc. of the 7th International Conference on Advanced Data Mining and Applications- Part II, pp. 292-303, Dec. 2011.

[2] N. Kawamae, H. Sakano, T. Yamada: "Personalized recommendation based on the personal innovator degree," Proc. of the third ACM conference on Recommender systems, RecSys '09, ACM, pp. 329-332, Oct. 2009.

[3] G. Adomavicius and A. Tuzhilin: "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Trans. on Knowledge and Data Engineering 17, pp. 734-749, Jun. 2005.

[4] J. Goldenberg, B. Libai and E. Muller: "Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth," Marketing Letters, pp. 211-223, Aug. 2001.

[5] D.D. Lee and H.S. Seung: "Learning the parts of objects by non-negative matrix factorization," Nature, Vol.401, No.6755, pp. 788-791, Oct. 1999.

*13 **Perplexity**: Technically "test-set perplexity," a quantitative evaluation measure primarily used to evaluate language models. Perplexity intuitively expresses the level of diversion of actual observed frequency from the probability generated by the model.

*14 **Force-directed layout**: A method of drawing graph data in an aesthetically pleasing way. Virtual force is assigned to the nodes and edges of the graph data, and then mechanically stable positioning is sought. This gives the edges more or less equal length, and enables edges to be positioned so that they intersect as little as possible.