International Journal of Grid and Distributed Computing
Vol. 11, No. 3 (2018), pp.99-108
http://dx.doi.org/10.14257/ijgdc.2018.11.3.09

Mobile Application Curation Service based on Big Data Platform

Jeong Hee Chi[1], Eui Seok Shim[2], Jeong Hee Hwang[3] and *Moon Sun Shin[4]

[1,2]*Department of Software, Konkuk University, Korea*
[3]*Department of Computer Science, Namsoul University, Korea*
[4]*Department of Computer Engineering, Konkuk University, Korea*
[1,2]*{jhchi,nasuk}@konkuk.ac.kr,* [3]*jhhwang@nsu.ac.kr,* [4]*msshin@kku.ac.kr*

Abstract

Popularization of smartphone allows many people to be able to share information, and easily get information they want. And then, app store of smartphone plays an important role to make users get information and make transactions they want easily. With the increase of the number of apps traded at the app store, it is difficult for users to find appropriate apps they want. Commonly, app store recommends apps users want through key words posted by users. But, this method has limits, and may not be satisfactory to users. This paper proposes a method where app store recommends apps to users by classifying apps used by users' friends by category and degree of satisfaction and recommending apps which are likely to be used. The experiment to test satisfaction of users showed that the method proposed in this paper increased the degree of satisfaction of users by over 21%.

Keywords: *App recommendation system, Personalization, Mobile curation service*

1. Introduction

Smartphone allows the user to process various works and multimedia information and others, using application software. application software working on smartphone is called app. Users find apps from app store. Initially, while there were not many apps, and users could easily find out apps they wanted, there was low diversity of apps. With the active use of smartphone, the number of apps has increased exponentially. The increased number of apps created a new problem such as difficulty of choosing proper apps. Users should invest time and expenses to decide proper apps among numerous similar apps. As ways to provide users better service, application market, that is, Google Play, App Store, and Facebook, etc. give recommendation service to users [1,2,3].

Recommendation system is information filtering system based on data analysis technology which is used in electronic commerce sites to help customers to search for goods they want to buy, by the methods of creating Top-N recommended goods list for specific user or predicting evaluation scores of the related user for the goods that may be recommended. Most of recommended systems are based on content-based (CB), or collaborative filtering (CF) algorithm. The latter is more commonly used, which creates recommended results based on the preference similarity between the users [6,10,12,14].

The recommendation by editors of Google Play Store is done by the method in which the application developer submits the app to the editors to recommend it by the subjective judgement standards of the developer, and editors examine them and recommend to users. They also provide users customized recommendation service using information about users. But, they recommend apps considering only limited information such as regions of users and the number of downloads. The Google Play Store exposes apps on the app list

Received (December 19, 2017), Review Result (March 10, 2018), Accepted (March 13, 2018)
* Corresponding Author

ISSN: 2005-4262 IJGDC

which paid larger advertisement fees first. Facebook provides application installed advertisements targeted according to gender, age, region, and personal interests of users. Currently, Facebook recommends game categories, but not others.

Such an app recommendation method is what editors recommend apps many people use. But, users are more interested in what those close to them are using than other many are using and tend to ask their friends characteristics of apps and effects of using them. They tend to choose apps those who are close to them use. This paper proposes an Appingpot system, method of evaluating and recommending apps based on analysis of app-using log users frequently use by using information on SNS-based friend relationship who share interests. By recommending apps not based on characteristics of apps and popularity of them to ordinary people, but on what those near users use, this method allows users to choose proper apps.

2. Related Works

Since 2008 when android application market launched, it has grown rapidly along with the spread of smartphone. And, not only the kinds and forms of service provided to users, and service types have gone through many changes in many fields. In contents service area as well, there are researches to provide quality service strategy to satisfy various kinds of demand from users. Various sensors installed on smart terminal and app surfing log history of a user is good information which can be used to grasp and predict patterns of the user. If such information is actively used, it is possible to provide personalized contents service. As attributes of app, there are function, cost, genre, completeness, and the number of downloads. The standard in cost aspect is divided into paid service and free service. Completeness to app is classified into five classes. Completeness is expressed through experiences on using apps. The number of downloads reflects popularization of apps.

Generally, the recommendation method [5-8] of the data mining [4] consists of the followings: content-based method, demographic method, and collaborative filtering method. Content-based method is word frequency method. Word frequency method is the method measuring mutual similarity by comparing content information of asked words and targeted objects. Such a method is suitable when objects which contain much text information. The method in which apps are recommended based on questions of users can also be called content-based method. But, as app is software expressed as digital information, it is difficult to extract apps proper to users only with similarity measurement on content-based method.

Demographic method is what predicts object preference of a specific user by calculating object preferences of other users using profiles of people such as job, gender, and age, *etc.*, [6, 9, 10, 11]. Collaborative filtering (CF) method extracts objects by discerning other users having similar patterns to the client recommendation will be given to and using information on object evaluation of them based on their preference histories [12-16]. To realize CF-based recommendation system, it is important to discern correctly users who are similar to the user who is the object of recommendation. In conventional collaborative filtering method, the similarity among users are calculated using quantitative information by using evaluation of them on goods and apps or the number of downloads.

Curation service algorithm the most frequently used is collaborative filtering algorithm, which consists of user-based one and item-based one. The most famous curation services, Amazon and Netflix built their recommendation systems using this method. This paper as well uses user-based collaborative filtering algorithm. In user-based recommendation system, what is important is how to construct user similarity measuring similarity among users. In measuring user similarity, usually Pearson correlation coefficient algorithm is used. Pearson correlation coefficient is the scale commonly used to examine relationship between tow variables.

Collaborative filtering algorithm predicts evaluation scores of the user who is the object of recommendation on specific goods through three stages. In the first stage, similarity between the user who is the object of recommendation and other users is calculated using Pearson correlation. In the second stage, based on similarity calculated in the first stage, *N* neighbors who are most similar to the user who is the object of recommendation are selected. In the third stage, based on evaluation scores of neighbors selected in the second stage, evaluation scores of the user who is the object of recommendation are predicted. Evaluation score $P_{x,i}$ of user x on goods *i* is calculated by the following equation (1).

$$P_{x,i} = \overline{r_x} + \sum_{z \in N}(r_{z,i} - \overline{r_z}) \cdot \frac{s_{x,z}}{\sum_{z \in N}|s_{x,z}|} \quad (1)$$

In equation (1), $\overline{r_x}$ is the average of evaluation scores of user x, and $s_{x,z}$ is the similarity between user x and neighbor user *z*. And *N* is the collection of the closest neighbors selected in the second stage, and *z* is the indicator of each neighbor.

Hadoop is Java-based open source framework which can make distributed processing of big data. Hadoop can store data in the HDFS (Haoop Distributed File System). This paper uses Hadoop 2.7.1, and uses spark, general purpose high performance distributed clustering platform. It is possible to add various modules on general purpose high performance distributed clustering platform that allows distributed multiple nodes to calculate. This paper realizes using ALS Algorithm [12, 13] based on collaborative filtering algorithm utilizing machine learning library, major function of spark.

3. Appingpot System Architecture

3.1. System Design

The system is composed of three modules: server module where data is pre-treated, big data module analyzing data, and client module giving service. Figure 1 shows the process diagram of Appingpot system. Table 1 describes the functions of modules on Appingpot system.

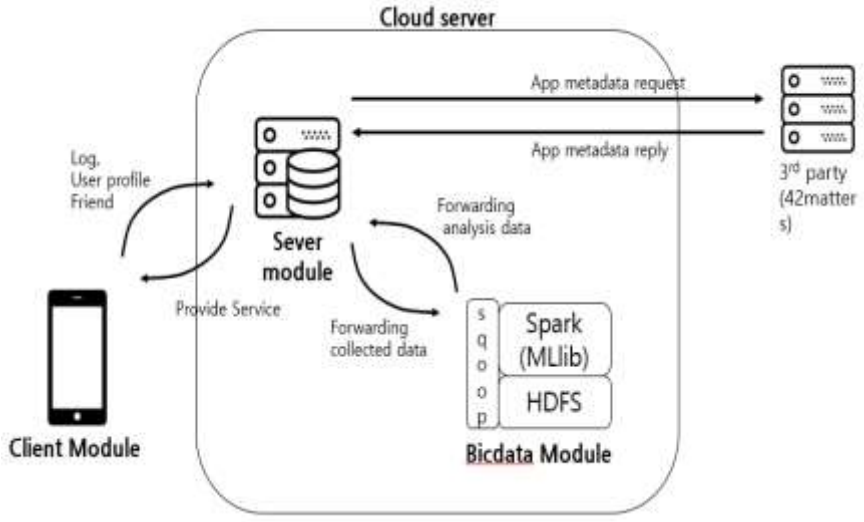

Figure 1. Overall Process of Appingpot System

Figure 2 shows server and client modules of our system proposed in this paper. Market info collector module composing server consists of 3rd party RESTful API Server(42matters) and market info collector server. It is the module which collects Google

Play market information and stores it in market DB using 3rd party service. The collector server collects and stores Google play market information once a day. If you demand Google play information in HTTP / GET, POST form, using 42matters, 3rd party service providing Google play market information in RESTful API type, it provides information by JSON. The market Info collector server collects market data provided by the 3rd party RESTful API Server once a day, and stores it in the market DB. Market module is composed of market info RESTful API server providing information to client with the RESTful API form, using Market DB where Google play market information is stored and Google play market information stored in market DB. Market Info Collector module stores Google Play once a day, and it not only provides basic market information using the data, but information on app chart it created for itself, which is important service.

Table 1. Definitions of Module Functions

Modules	Description
Android Client Module	It collects membership information, Facebook friends information, and app using log data, and send them to servers. It provides recommendation service analyzed through data to users.
Server Module	It is server module composed of NodeJS. It manages membership information and log data DB, and recom mends apps acquired from big data to users.
3rd party (42matters)	Using additional information on recommended apps th rough 42matters, Android market information API co mpany, it provides additional information on recomme nded results.
Big Data Analysis Module	It analyzes log data and recommends by doing pre-pro cessing and standardization process of log data.

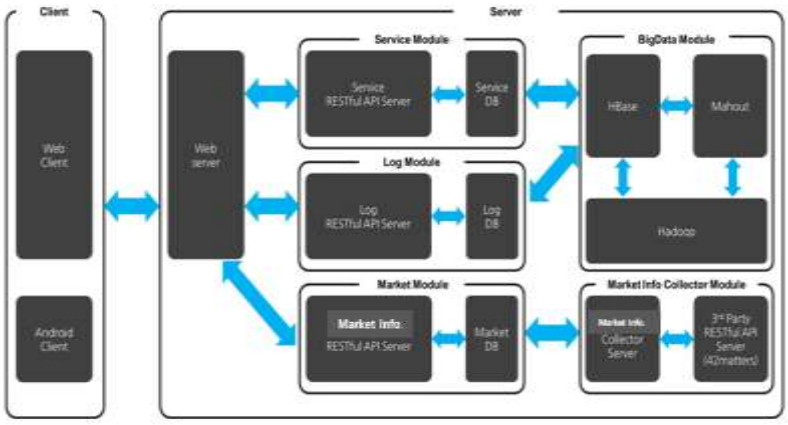

Figure 2. Appingpot System Flow Diagram

Log Module stores log information on app using patterns of clients and provides such log information to client or other RESTful API servers. Log RESTful API Server provides service on log information to client or other Restful API servers by RESTful API forms. Service Module provides app recommendation service and additional service. It is composed of Service RESTful API Server which provides service with RESTful API forms and ServiceDB storing client information.

BigData Module analyzes big data through Hadoop based on market information, log information, and client information. It consists of Hbase, Mahout, and Hadoop. Hbase is

the system which stores and processes data as big tables to facilitates big data distribution processing for real-time analysis of big data. It stores user log information and Google play information to facilitate distribution processing. Mahout works on Apache Hadoop using MapReduce to classify and define data and do collaborative filtering, and recommend apps using app list and app log information stored in LogDB and ServiceDB. Hadoop does distribution processing of big data needed to analyze membership information data and user log analysis and analyze its own analytic chart.

Clients in client module consists of Android client who is common user and web clients who is manager. Android client provides applications to common users with recommendation information, application information, and Android market information, *etc*. And, Web client is used by manager, and provides Google play market information, user application recommendation information, user information, and service operation information, *etc*.

3.2. Android App Service Modules

Figure 3 shows the class diagram of android app service modules. Tracker class of Android module measures app use amount and calculates using time and frequency per hour. TrackerDAO class stores app use amount per hour in mobile DB. ForegroundEvent brings all the app use event information, and stores app use amount and use time per hour. RestClient manages client service to use REST API.

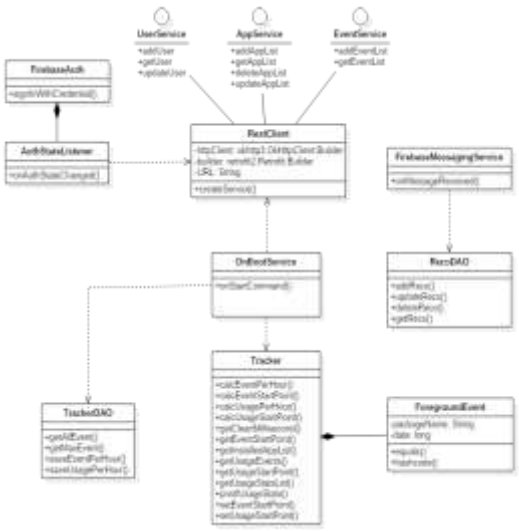

Figure 3. Class Diagram of Android App Service Modules

System functions are as follows. Login function is divided into two kinds: users who join after joining Facebook; nonmembers. When a user login for the first time, he or she can input information on preferred categories by registering on them. If a user login through Facebook, it provides the user with information on Facebook friends who use the recommended app when it recommends the user an app. If a user login as nonmember, the user cannot use information on friends. By providing the beginning user who uses it for the first time after joining Facebook with simple tutorial, it enhances the user's understanding on how to use the program.

Using ALS algorithm utilizing collaborative filtering to examine application use log records of user, it recommends apps to the user. When recommended candidate set is completed through algorithm, weights are imposed on interested categories and Facebook friends who have similar app list, and the final application recommendation dataset is completed. By this method, optimal application to the user is recommended. And, by

analyzing app use log records of the user, it notifies the user of apps the user rarely uses and provides the user with the function of deleting them. And, by analyzing apps existing users frequently use per category, it notifies the user of recommendable or popular apps, if there are any.

An important function provided to common users is curation function. It analyzes applications a user uses per category, and analyzes Android market, and recommends the user the most suitable application. If there is an app more popular or recommendable than existing app, it notifies the user of it. And, it creates a chart by analyzing apps in the application market and provides the chart to the user. It provides the user with market information Google Play has provided category, the number of downloads, promotion video information and self-created Android market information (app information SNS friends have downloaded, the age of users who use it, and real-time popular apps).

For the management service for apps currently installed, it visualizes app use patterns of a user. It informs the time to use it, frequency of using it, use amount per category, allowing the user to use apps efficiently. It also informs the user of the list of rarely used apps, and the user can delete them. In addition, it also informs the user of feed information such as update news, newspaper article on related app of the app the user chose.

Major functions provided to the manager are as follows. It provides manager with the information that Google play provides nationality, category, charged or free, date, and number of downloads, etc. and detailed information on specific app such as time when users usually use, gender, age, and relations with other apps, etc. And, by analyzing logs of users, it provides information on app installment inflow channels. It also provides users with recommendation system information, recommendation method statistics, click rate and installment rate after app recommendation, and recommendation log, etc. It also provides basic membership information, whole use application use pattern analysis and statistics information on how much the user uses paid service.

4. Implementation and Analysis

To realize the app recommendation system proposed in this paper, it is necessary to store information on app use and friends of the user in MySQL. And, using this, it needs to pre-process based on preference scores, and transmits to Hadoop Distribution File System (HDFS) taking advantage of Sqoop which allows data transmitted between Hadoop and relationship-type DB. Data collected for realization collects installed app information list, number of app use per hour, time of app use per hour, list of Facebook friends, and user's interest categories, through Appingpot log module. In the collected data, number of app use per hour and time of app use per hour are important. These two kinds of information are calculated as average use time of one operation of app in use server module. And, through pre-processing procedure, preference scores are calculated and stored in DB. Log data on app use frequency and app using time are stored in the DB table, which allows one to calculate average use amount per one app use. However, since this value has wide boundary, it goes through normalization. The more the user uses it, the higher the preference score goes. The average usage time per one use is calculated by following equation (2).

$$Average\ usage\ time = \frac{Total\ usage\ time}{Number\ of\ times\ used} \tag{2}$$

In the analytic process, it is possible to collect information on friends through SNS login and enhance accuracy of recommendation with friend relationship. And, with photos of friends, it is possible to raise reliability of the user, and enhance the accuracy of recommendation by receiving information on interests of the user. What should be considered for realization and what needs to be complemented and improved are as

follows. First, if information acquired from the user is not sufficient, accuracy of recommendation can go down. To compensate this problem, it is necessary to raise the accuracy of recommendation by using not only application list, but SNS account information, and interest tag information. Second, it is necessary to consider leakage of security of personal data of the user and analyzed data. To do this, it is necessary to use cloud server and, sometimes, paid service. Third, users may not like revelation of information about me to other friends, because SNS information is used. Therefore, it is necessary to make the user to set the degree of exposure of personal information.

Data analytic process is composed of four steps. The explanation of each step is as follows.

Step 1: Data-sending step. Using Sqoop, it sends pre-processed data located in RDBMS to HDFS. Pre-processed data is established as DB. Using the module called Sqoop designed to send data among Hadooop Echo Systems, data located in MySQL moves to Hadooop HDFS.

Step 2: the step creating the first candidate set with collaborative filtering. It selects contents preferred by many users and which is similar to what the user prefers as the first candidate set. Using MLlib (machine learning library) of Apache Spark and Alternating Least Squares Recommender Algorithm (ALS Algorithm), it generates he first candidate set depending on preference patterns of similar users. It uses MLlib provided by Spark. And, it uses collaborative filtering-based ALS algorithm. Using information on users who have use patterns similar to that of the user, it calculates what preference scores the related user has on a new item, it creates the first candidate set.

Step 3: the step where it generates the second candidate set using Facebook friends. By identifying Facebook friends, and imposing weights on the list, and creating the second candidate set. The second candidate set uses the information on Facebook friends. Analyzing the similarity of the app list of Facebook friends who use the first candidate set with Euclidian similarity distance, it imposes weights on the list. By arranging them with expected scores through expected score plus average similarity, it creates the second candidate set. The figure 4 shows how expected scores are calculated.

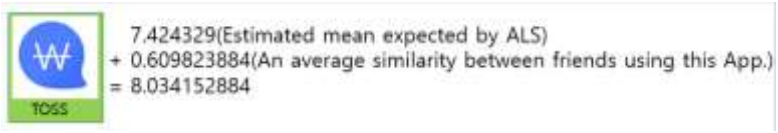

Figure 4. An Example of Android App Recommendation Score Calculation

Figure 5. Generation Process of Recommended Apps

Step 4: the final step creating the final candidate set using interest tags. The final candidate set uses interest tags. When the second candidate set is created, it collects meta information of the related app through API of 43matters company providing application meta information. It collects titles, icon information, market URL, and category information needed to give recommendation. Also, if the list matches categories preferred by the user, weights are imposed, creating the final candidate set. Using this, it provides service to users. Figure 5 shows generation process of recommended apps.

Currently, Netflix is a typical curation service. Customer satisfaction with recommendation provided by Netflix is known to be around 80%. We did experiment to 50 users with the system proposed in this paper where we asked them to evaluate satisfaction with the recommended apps for three months after apps are installed. Test results showed that recommendation satisfaction increased by about 21% compared with the case none was recommended. The use of Facebook friends and interest tags affected the improvement of recommendation satisfaction.

The proposed system can be applied to be expanded as service providing app marketing and OpenAPI. The kinds of information the system collects are various: age, gender of users and others. These kinds of information can be applied in grasping trends of app market and planning and marketing areas. Second, it can be used for service providing OpenAPI. Nowadays, there are many OpenAPI services which can use various services. The sources analyzed in this system and recommended service can be used as service which can be provided to more people through OpenAPI services.

5. Conclusions

Smartphone app store which enables users to download application programs they want allows them to conveniently acquire information and be able to do transactions easily. But, with the increase of the number of apps traded at the app store, it is hard to find suitable apps users want. Therefore, users are interested in what apps their friends are using and tend to ask the characteristics and effects of apps their friends use. Accordingly, they tend to use apps their neighbors favor. This paper proposes the Appingpot system which evaluates apps based on app use log analysis favored by other users by using information on SNS-based friends of users who share interests and recommend those apps to users. The experiment result shows that the recommendation method proposed in this paper provides information directly useful to users, because it recommends apps not based on general characteristics of them and popularity among people, but on use of apps by friends of users.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A3B03033944)

References

[1] S. K. Ray and S. Singh, “Blog content based recommendation framework using WordNet and multiple Ontologies”, 2010 International Conference on Computer Information Systems and Industrial Management Applications, **(2010)**, pp.432-437.
[2] U. Shardanand and P. Maes, “Social information filtering: Algorithms for automating ‘word of mouth’”, Proceedings of the Conference on Human Factors in Computing Systems **(1995)**, pp.210-217.
[3] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting web sites," Proceedings of the 13th National Conference on Artificial Intelligence, vol.1, **(1996)**, pp.54-61.
[4] J. Han and M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann, **(2001)**.
[5] L. Sebastia, I. Garcia, E. Onaindia, and C. Guzman, “e-Tourism: A Tourist Recommendation and Planning Application”, International Journal on Artificial Intelligence Tools, vol.18, no.5, **(2008)**, pp.717-738.

[6] C. Stiller, F. Ross, and C. Ament, “Demographic recommendations for WEITBLICK, an assistance system for elderly”, International Symposium on Communications and Information Technologies, **(2010)**, pp.406-411.
[7] T. Chen, W. Han, H. Wang, Y. Zhou, B. Xu, and B. Zang, “Content Recommendation System Based on Private Dynamic User Profile”, International Conference on Machine Learning and Cybernetics, **(2007)**, pp.2112-2118.
[8] C. Jian, Y. Jian and H. Jin, “Automatic content-based recommendation in e-commerce”, The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, **(2005)**, pp.748-753.
[9] T. Chen and L. He, “Collaborative Filtering Based on Demographic Attribute Vector”, International Conference on Future Computer and Communication, **(2009)**, pp.225-229.
[10] B. Krulwich, "Lifestyle Finder: Intelligent user profiling using large-scale demographic data," Artificial Intelligence Magazine, Vol.18, No.2, **(1997)**, pp..
[11] M. Pazzani, “A Framework for Collaborative, Content-Based, and Demographic Filtering”, Artificial Intelligence Review, **(1999)**, pp.393-408.
[12] Y. Hu, Y. Koren and C. Volinsky, “Collaborative Filtering for the Implicit Feedback Datasets”, IEEE International Conference on Data Mining, **(2008)**, pp.263-272.
[13] Y. Zhou, D. Wilkinson, R. Schreiber and R. Pan, “Large-scale Parallel Collaborative Filtering for the Netflix Prize”, Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management, **(2008)**, pp.337 - 348.
[14] M. N. Uddin, J. Shrestha, and G. Jo, “Enhanced Content-Based Filtering Using Diverse Collaborative Prediction for Movie Recommendation”, 2009 First Asian Conference on Intelligent Information and Database Systems, **(2009)**, pp.132-137.
[15] Z. Zhang, D. Zhang, and J. Lai, “urCF: User Review Enhanced Collaborative Filtering”, Proceedings of the 20th Americas Conference on Information Systems, (2014), pp.1-11.
[16] Y. Wang, Y. Liu, and X. Yu, “Collaborative Filtering with Aspect-Based Opinion Mining: A Tensor Factorization Approach”, Proceedings of IEEE 12th International Conference on Data Mining, **(2012)**, pp.1152～1157.

Authors

Jeonghee Chi, she received Ph.D degrees from Chungbuk National University in 2006 in computer science. In 2007 she joined the faculty of Konkuk University, where she is now an assistant professor. Her research interests include IoT, stream data mining, machine learning and big data analysis.

Euiseok Shim, he is a undergraduate student of the department of software, Konkuk University. He is interested in IoT and big data analysis.

Jeonghee Hwang, she received Ph.D degrees from Chungbuk National University in 2005 in computer science. In 2006 joined the faculty of Namseoul University, where she is now an assistant professor. Her research interests include data mining, semantic web, ubiquitous computing and big data processing.

Moonsun Shin, she received Ph.D degrees from Chungbuk National University in 2004 in computer science. In 2005 she joined the faculty of Konkuk University, where she is now an associate professor. Her research interests include IoT, ICT convergence, context awareness, machine learning and big data analysis.