Neurocomputing 326–327 (2019) 100–109

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hierarchical modular Bayesian networks for low-power context-aware smartphone

Sung-Bae Cho*, Jae-Min Yu

Department of Computer Science, Yonsei University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 31 May 2016
Revised 12 December 2016
Accepted 29 January 2017
Available online 21 September 2017

Keywords:
Bayesian networks
Context-aware service
Battery saving
Smartphone

ABSTRACT

Nowadays, smartphone has a tremendous number of applications using sensors and devices for several applications such as healthcare and game. However, serious consideration to increase the duration of battery use of phone is required because of the limited battery capacity. In this paper, we propose a hybrid system to increase the longevity of phone with hierarchical modular Bayesian networks that recognize the user's contexts, and device management rules that infer the unnecessary devices in smartphone. Inferring the user's contexts with sensor data and considering the device status, the context inferred and user's tendency, we determine the superfluous devices that are consuming the battery as dispensable. The experiments with the real log data collected from 28 people for 6 months verify that the proposed system performs the accuracy of 85.68% and the reduction of battery consumption of about 6%.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

According to the technical document of Qualcomm in 2013, the most serious concern of mobile users is the duration of battery [1,2]. Most of the users have to bring spare battery together in spite of the improvement of battery capacity. The data collected from the sensors and devices of smartphone can be to infer the user's context and develop applications with appropriate services. However, as the users demand intensely for long-time use, it encourages the active research on increasing the battery-use time for smartphone.

In this paper, we propose a low-power device management system for smartphone using hierarchical Bayesian networks that can infer user's context and unnecessary devices. The proposed system infers the user's context using Bayesian networks. Considering the user's context, the tendency inferred and the status of devices, we can infer the superfluous devices that can consume battery as dispensable. As a result, the proposed system can improve the duration of battery. In order to reveal the usefulness of the proposed system, we evaluated the accuracy of identifying the unnecessary devices by real data collected from 28 people, and compared it with the competitive methods.

The rest of the paper is organized as follows. Section 2 introduces the related work regarding the mobile applications on device management. Section 3 presents the hierarchical modular Bayesian networks to reduce the battery consumption of smartphone. In Section 4, we show the experiments and the results to evaluate the proposed system. Finally, Section 5 summarizes this paper and presents the future works.

2. Related works

Mohammad recognized user's context in smartphone, and proposed a method to manage the devices. In this research, to reduce the complexity of context-awareness, he incorporated fuzzy inference [13]. Zhuang adjusted update-time-gap to gain location information [10]. Herrman proposed a low-power system that can adjust the state of sensor devices according to context [12]. Xu proposed a technique by controlling display brightness level and GPS sampling rate on smartphones [22]. Previous works defined static situation for low-power platform [6,9,11], and adjusted limited-sensor devices as shown in Table 1.

The previous applications for low-power system did not exploit the users' use pattern, and had a problem that they cannot configure the setting directly [14–17]. Applications of considering the use pattern have problem of cold-start that takes a long time to recognize the pattern and has few data samples. In order to work out this problem, we propose a device management system that can adjust appropriate devices for user through context inference. The proposed system supports the automatic adjustment of battery-saving mode in accordance with each situation. It can improve the accuracy of situation inference through learning of use patterns.

Table 2 shows the commercial low-power device management applications. Battery Guru developed by Qualcomm can grasp use-pattern and optimize the device function of smartphone. This ap-

* Corresponding author.
E-mail address: sbcho@yonsei.ac.kr (S.-B. Cho).

https://doi.org/10.1016/j.neucom.2017.01.124
0925-2312/© 2017 Elsevier B.V. All rights reserved.

Table 1.
Related works on low-power platform. Notice that O means the inclusion of the property, whereas X means the non-inclusion.

Researcher	Function		Description
	Context awareness	Device adjustment	
Xu et al. (2014) [22]	O	GPS, Brightness	Markov decision process based on rules and action function
Mohammad et al. (2013) [13]	O	X	Fuzzy inference model of low computational complexity considering context-awareness of user in smartphone
Herrmann et al. (2012) [12]	X	Sensor devices	Low-power system adjusting power consumption of sensor device according to user's context
Weiss et al. (2011) [8]	O	X	Analysis of correlation of context-awareness of users using accelerometer sensor in smartphone for 70 people
Zhuang et al. (2010) [10]	X	GPS	Adjustment of update-time-gap of location information about user's static state
Bettini et al. (2007) [7]	X	Power consumption	Investigated power consumption of built-in sensor such as accelerometer, microphone, GPS, Wi-Fi, and Bluetooth

Table 2.
Commercial device management applications. Notice that O means the inclusion of the property, whereas X means the non-inclusion.

Researcher	No. of saving modes	Configuration setting	Use pattern using
Battery Guru	2	O	O
Battery Doctor	3	X	X
King of Energy Saving	3	O	X
2 Battery	2	O	X
Green Power	2	O	X
MX Battery Saver	4	X	O
DU Battery Saver	3	O	X
Battery Saver 2	2	X	X

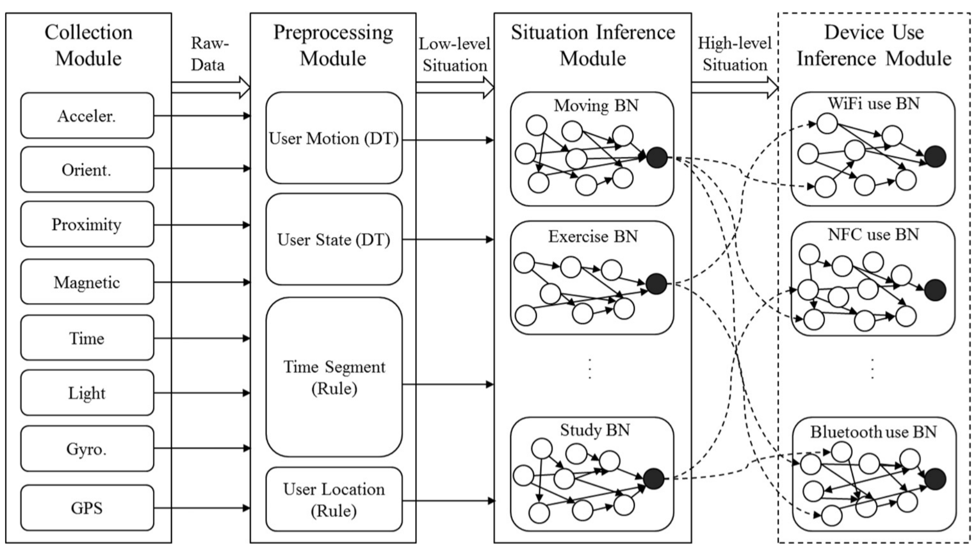

Fig. 1. The overall system structure.

plication leads to decreasing battery consumption. However, these previous applications have a problem that takes a long time to recognize pattern or relies only on user's configuration.

3. The proposed system

In this paper, we analyze the correlation between the situation and tendency of user and the devices to develop a low-power platform by the literature survey and the data collected. At first, we preprocess the sensor data by decision tree and rules for the input to Bayesian networks, because it is relatively easy to implement decision tree and rules that have advantage of high accuracy and fast processing for the information like acceleration data [3]. After we identify the state of user's movement (walk, run, and stop) using acceleration sensor, and user's posture (sit, stand, and lie) using orientation sensor, we distinguish user's location of indoor and outdoor using the rules with the information of GPS (We should replace the GPS with the alternatives because it consumes the large amount of battery).

After analyzing the correlation among the data collected, we develop the Bayesian networks to infer the user's situation. We have used the modular Bayesian networks with tree structure using the low-level data preprocessed. Bayesian networks are a good model for handling uncertain input in smartphone environment. As they are constructed as tree structure, they can infer the situation with low calculation complexity compared to the conventional Bayesian network. It has an effect on the low power consumption of CPU. As network can be a powerful model through

Table 3.
Feature values used for decision tree.

Feature value	Description
sum_accX	Summation of X-axis acceleration
sum_accY	Summation of Y-axis acceleration
sum_accZ	Summation of Z-axis acceleration
std_accX	Standard deviation of X-axis acceleration
std_accY	Standard deviation of Y-axis acceleration
std_accZ	Standard deviation of Z-axis acceleration
sum_orientation	Summation of orientation
sum_pitch	Summation of pitch
sum_roll	Summation of roll
std_orientation	Standard deviation of orientation
std_pitch	Standard deviation of pitch
std_roll	Standard deviation of roll
sum_magX	Summation of X-axis magnetic field
sum_magY	Summation of Y-axis magnetic field
sum_magZ	Summation of Z-axis magnetic field
std_magX	Standard deviation of X-axis magnetic field
std_magY	Standard deviation of Y-axis magnetic field
std_magZ	Standard deviation of Z-axis magnetic field
SMA	Signal magnitude area

Table 4.
The result of sensor data preprocessed.

Method	Sensor types	Use patterns
Decision tree	Acceleration, direction	Walk, run, stop
	Magnetic, Gyro	Sit, stand, lie
Rules	Light	Very bright, bright, normal, dim, very dim
	Time	Morning, afternoon, evening, dawn
	GPS	School, home, library, theatre, cafeteria
	Battery amount	Low, normal, high

cumulated massive data learning, researchers are exploiting learning algorithms. In order to run the decision tree in mobile environment in real time, we implement it in C language on Android platform through NDK. The code implemented in C might run faster because it does not use Java virtual machine (see Fig. 1).

As sensor data have significant variation according to user's state, we have to classify the time-series sensor data into user's movements. To detect the user's movement state, we extract the feature values as input of decision tree first. The data used in decision tree come from acceleration, magnetic and orientation sensors. We use the four feature values for each sensor, calculated by the following equations.

$$sum\_X = \sum_{i=1}^{N} |x_i - x_{i-1}| \tag{1}$$

$$mean\_X = \frac{\sum_{i=1}^{N} \sqrt{(x_{i+1} - x_i)^2}}{N} \tag{2}$$

$$SMA\_X = \sum_{1}^{N} (|x_i|) + (|y_i|) + (|z_i|) \tag{3}$$

$$std\_X = \sqrt{\left(\sum \sqrt{(x_{i+1} - x_i)^2} - mean\_X\right)^2} \tag{4}$$

Here, X represents sensor, x means the value at present time of i, and N means the total amount of data in a window. Table 3 shows the total feature values used for the decision tree.

The proposed system consists of data collection module, preprocessing module, situation inference module, and device-use inference module. Data collection module collects mobile sensor data and the status of device. Table 4 shows the sensor types and use patterns of the decision tree and rules for preprocessing module. Situation inference module infers the user's context using the input data preprocessed. Considering the tendency of user, battery status, and inferred situation, device-use inference module can infer superfluous devices and adjust them accordingly.

3.1. Modular Bayesian networks

Bayesian networks are models that express large probability distributions with relatively small cost in statistical mechanics. The structure is a directed acyclic graph (DAG) that represents the link relationship of each node and includes conditional probability table (CPT). It consists of two components: structure and parameters. The network can identify the structure and parameters through learning algorithms using real world data as well as constructing it with expert's domain knowledge.

Modular Bayesian networks (MBN) is an extended version of Bayesian network [19]. MBN has the basic BN's features such as *d*-separation and increasing complexity as the growing number of parent's nodes. Besides, it is possible to apply BN's inference or learning algorithms to MBN without modification. In MBN, however, it is more difficult to keep dependencies between variables, especially inter-modular causality due to its modular property. In addition, a clever method is required in inference since updating the whole MBN given small change of observation would take considerable resources and time. To settle down these problems, the system uses virtual linking and selective inference for MBN. Inter-modular causality is preserved by virtual link between modules, and only the necessary modules are taken into account in inference process. Before delivering detailed method, clear definition and important feature of MBN are described here. MBN works based on BN and consists of multiple BN units (BN modules) which are connected with other units according to their causality relationships. BN modules and MBN are defined as follows.

Definition 1 (BN module). A BN module $\psi_i = (G_i, P_i)$ is a Bayesian network represented by $G_i = (V_{\psi_i}, P_{\psi i})$ where $V_{\psi i}$ are the set of random variables belonging to G_i and $E_{\psi i} = (X, Y)$ are directed edges from X to Y. X and Y are random variables belonging to $V_{\psi i}$. Every BN module is modeled by the subset $V_{\psi i}$ of all set of random variables U that can be observed in the BN model. The conditional probability table $P(V_{\psi i})$ in BN modules has the same definition with $P(V_i)$. The BN module is a subgraph of the monolithic BN [20].

Definition 2 (modular BN). MBN Ω consists of a 2-tuple (ψ, R) where ψ represents all set of BN modules in Ω and R indicates the causality among BN modules. Let two BN modules be defined as $\psi_i = ((\mathrm{V_i},\ E_i),\ \mathrm{P_i})$ and $\psi_i = ((\mathrm{V}_j,\ E_j),\ \mathrm{P}_j)$, and have the influence on each other. Then, a link $\mathrm{R} = \{\langle \Psi_i, \Psi_j \rangle \mid i \neq j,\ V_i \cap V_j \neq \emptyset\}$ is created and able to affect or be affected by defining a sharing node.

Definition 3 (design criteria of MBN). There are four design criteria of the MBN: (1) A pair of modules is non-symmetric. In other words, $\langle \psi_i,\ \psi_j \rangle$ is not equal to $\langle \psi_j,\ \psi_i \rangle$. (2) MBN does not allow a circular loop to its own. (3) A pair of BN modules has at least one shared variable. (4) MBN can be reelected the mutual causality between the modules. That is, it is possible to change the input and output modules depending on situation.

When a module is updated we can restrict the range of modules to be re-inferred by the inter-modular relation. Due to this reason, modular *d*-separation needs to be checked dynamically. This is done by updating module selector in the proposed MBN system [19]. This modular Bayesian network was applied in landmark detection with mobile devices and distributed multiple sensor networks [21]. Fig. 2 shows the modular Bayesian network designed for inference of device status.

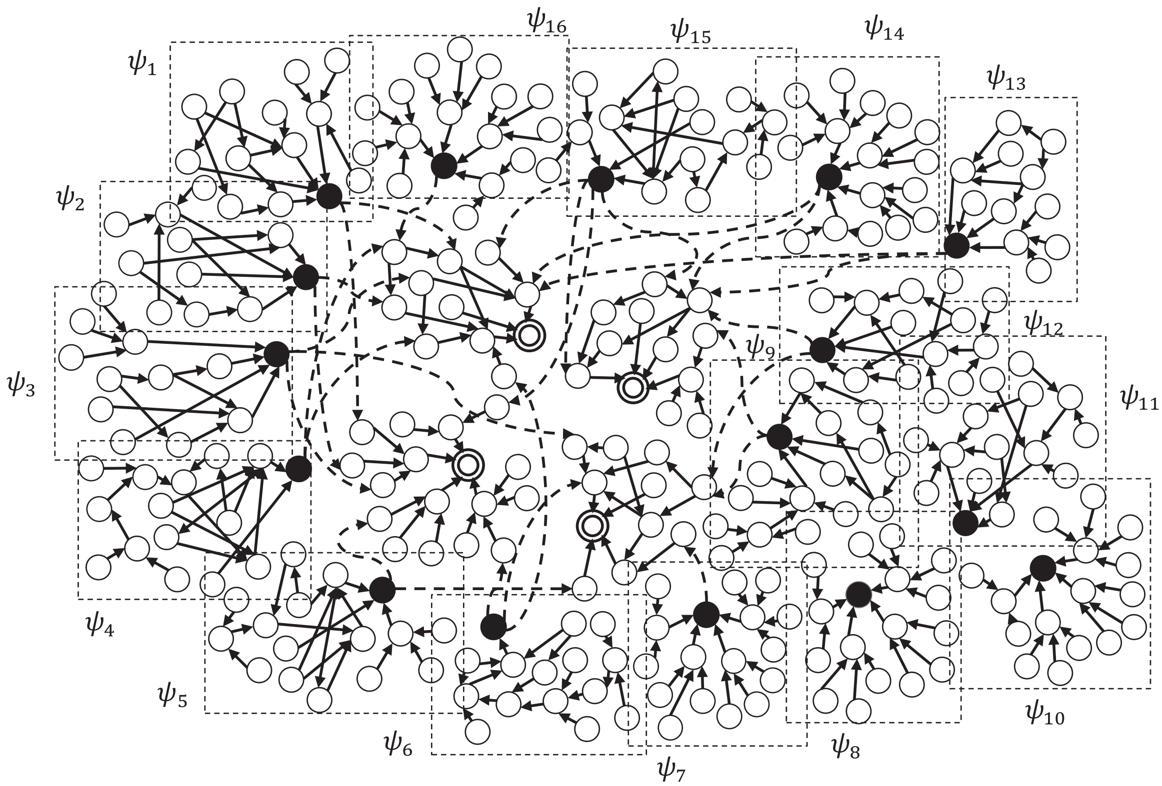

Fig. 2. Modular Bayesian network for inference of device status.

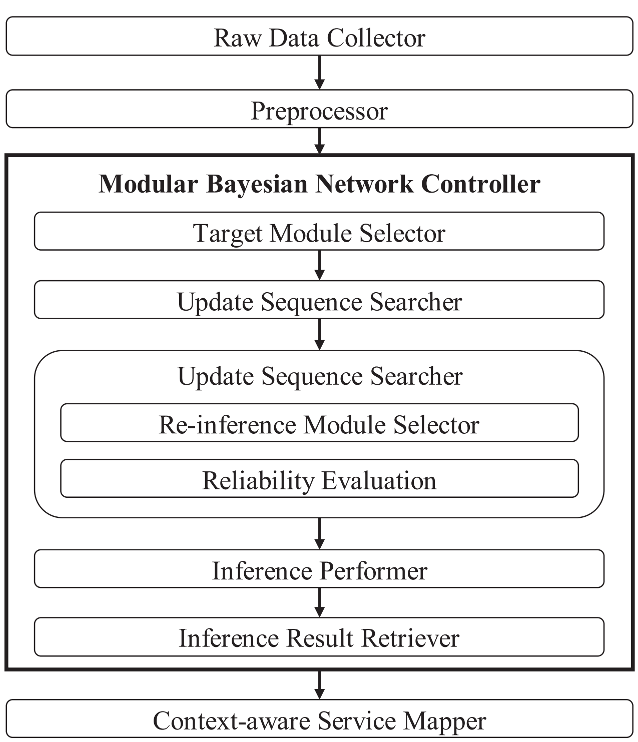

Fig. 3. Modular Bayesian network system with functional components.

3.2. Inference with modular Bayesian networks

The inference process of MBN consists of multiple steps as shown in Fig. 3 [19]. The raw data collector obtains continuous sensor data and log data from the smartphone (e.g., accelerometer, gyroscope, time, schedule, and so on). The data are sent to the preprocessor for discretization using algorithms such as decision tree, naïve Bayes classifier, etc. The data preprocessing aims to reduce the time complexity of the BN modules. The number of states of the input value increases the size of the CPT. MBN infers the context, and finally the context-aware service mapper selects the optimal services in this inferred context.

The MBN selects the target nodes needed to know in accordance with the collected contextual information. A target module contains a target node. When the target module is selected, the system then selects the modules that need re-inference. When the system conducts selective inference, it selects modules that possibly affect target modules and change the posteriori probability of the modules. This system stores the previous module's evidence set because the posteriori probability of a module depends on the evidence set.

3.3. Hierarchical probabilistic model

In a previous work [3], we recognized the user's situation in low power using linear inference. We recognized the situation of rest, sleep, dining, exercise, work, shopping and lesson using the proposed modular situation inference. According to the criteria of situation classification of National Statistical Office in Korea [4], we define eight situations. Unnecessary device could be turned off, because this might consume battery. For an example, in outdoor, if Wi-Fi device is turned on, we do not use Wi-Fi. However, that device always can try to search access point or Wi-Fi connection nearby. The inference technology proposed in this paper figures out the superfluous state of each device hierarchically considering the user's tendency and present mobile state. This method reuses the result of user situation inference as evidence value. We use the user's tendency from Big-five tendency model proposed by McCrae and John [5]. Table 5 is the definition of I/O of probability model of each device.

Table 5.
Input and output of Bayesian network of inferring unnecessary devices.

Classification	Type	Description
Input	Sleeping	We consider co-relationship between the result of situation inference of user and device use based on the situation classification of National Statistical Office.
	Dining	
	Work	
	Lesson	
	Watching	
	Exercising	
	Moving	
	Rest	
	Battery state	Considering usable device or not via battery state
	Screen state	Considering the state of screen be turned on or off
	Extroversion	We consider co-relationship with the tendency of user based on Big-five model.
	Openness	
	Congeniality	
	Sincerity	
	Faithfulness	
Output	Device	Wi-Fi, GPS, Bluetooth, Data synchronized device

Input:
User situation, tendency of user, mobile state
Output:
State of devices unnecessary
Device = $\{d_1, d_2, \ldots, d_i\}$
While (an inference module can be selected according to the state of devices) {
Configure the input value for each module;
Calculate the probability linearly;
Decide the state of superfluous device based on critical value;
}

Fig. 4. Inference algorithm of unused devices.

3.4. Inference and management of unnecessary devices

We infer the unnecessary device based on the linear inference algorithm by Das [18] considering only coincidence of state value about cause-and-effect relationship of output value as the following Eq. (5).

$$P(S) = \sum_{j=1}^{n} w_j p\big(S|Comp\big(I_j = i_j\big)\big) \tag{5}$$

where $P(S)$ is the probability of situation S. I_j is the input of node j, and i_j is the input value for node j. $Comp(I_j = i_j)$ is the value of conditional probability table on the coincidence situation of node j. w_j means the influence of node j to the final situation.

Input:
User situation, tendency of user, mobile state
Output:
State of devices unnecessary
Device = $\{d_1, d_2, \ldots, d_i\}$
While (an inference module can be selected according to the state of devices) {
Configure the input value for each module;
Calculate the probability linearly;
Decide the state of superfluous device based on critical value;
}

Fig. 4 shows the inference algorithm to find unnecessary devices. First, we choose the inference module in order to understand the state of device d_i. We configure the evidence value of input node of smartphone and the tendency and situation of user needed for inference module, and calculate probability value of unnecessary node. Next, we calculate conditional probability value and intermediate node. Finally, calculate probability value of result node that means the state of unnecessary device, and decide the unnecessary state of device based on a threshold. This process is repeated for device management inference module. Through the repeated result, we carry out device management.

Fig. 5 shows the graphic user interface of device management for each situation. If user runs the application located in background screen on smartphone, user can confirm devices that can be adjusted by user. The GUI is designed to configure battery-saving level, and user can confirm to configure the setting and working system.

4. Experimental results

4.1. Experiment environment

The experiments were conducted with the 6985 real log data collected from 28 people for 6 months. 16 situations inferred include sleeping, eating, work, lesson, housework, watching, exercise, study and others. Sensors used are GPS, Bluetooth, Wi-Fi, and synchronization. Samsung Galaxy S4 is used for the data collection and application software running. Even though the application was developed at the desktop computer with Intel Core i7-2600L, 16.0 GB RAM, and Windows 7, all the experiments and computations are performed on the smartphone.

To show the usefulness of the developed application in low power environment, we designed a scenario which is effective for battery saving and utilizing device management application. The scenario aims to show the effect of device management application through situation inference and alteration of state of device in addition to situation and tendency of user. Figs. 6 and 7 show the difference between with and without the device management application through the scenario. In this figure, the scenario shows a possibility to reduce battery consumption. When the user takes a lesson, Wi-Fi, Bluetooth and GPS get automatically turned off at 8:50 (Fig. 7). Similarly, Wi-Fi module is turned off when user is studying in library at 13:25.

To verify the proposed system, we conduct the quantitative experiments in four parts. First, we measure the classification accuracy (from 10-fold cross validation) of movement and transportation, and compare with the competitive classifiers to confirm the performance of decision tree. Second, we measure the inference accuracy of situation with the 16 modular Bayesian networks. Third, we measure the inference accuracy of unnecessary devices based on the situation inferred by the previous experiment. Finally,

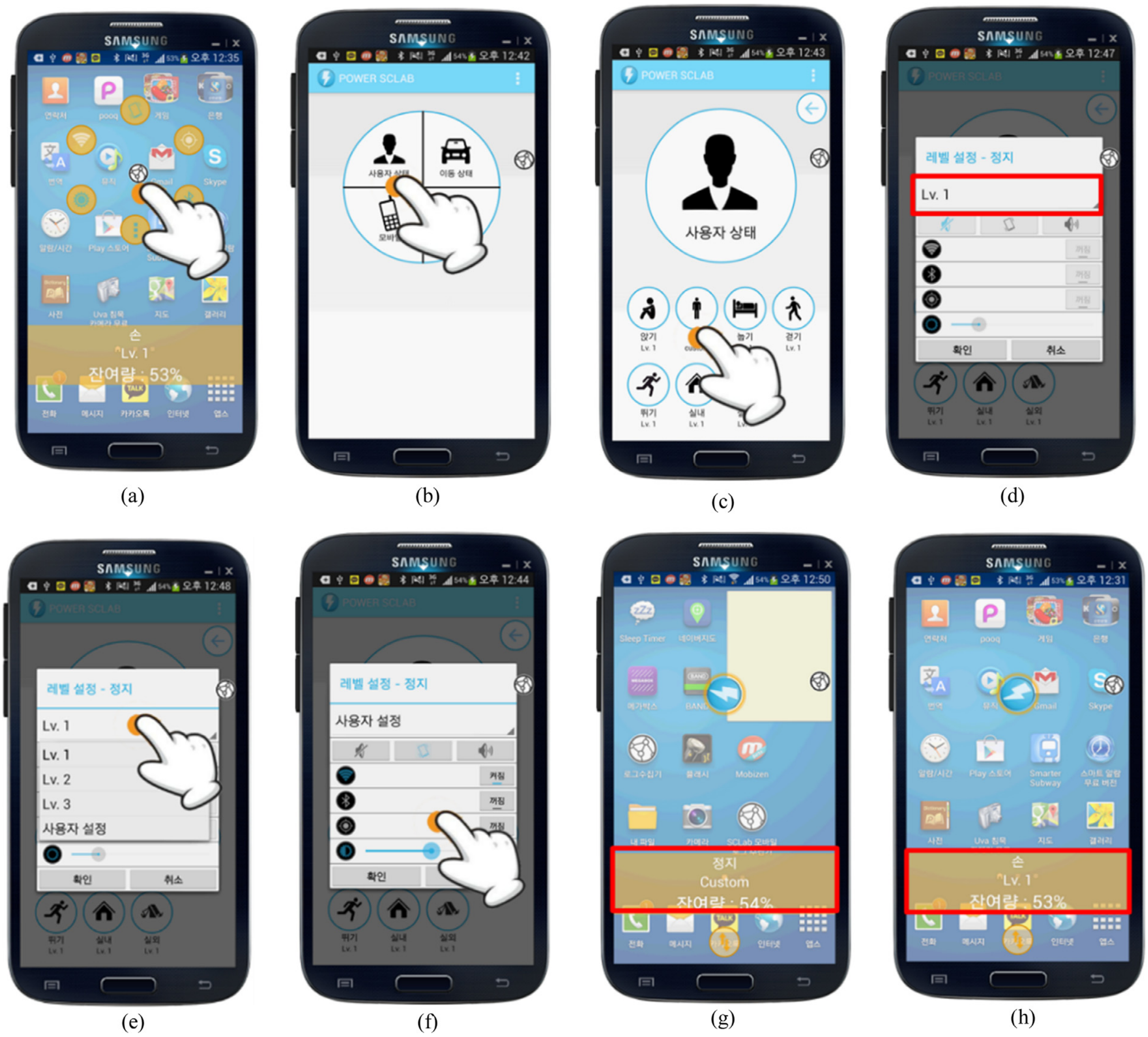

Fig. 5. The GUI's of the device management application. A user can start the application by clicking the circular buttons (a), select the appropriate status to input the values for the situation (b, c), and enter the levels and device status (d, e, f). The application displays the current situation inferred and the remaining battery (g and h).

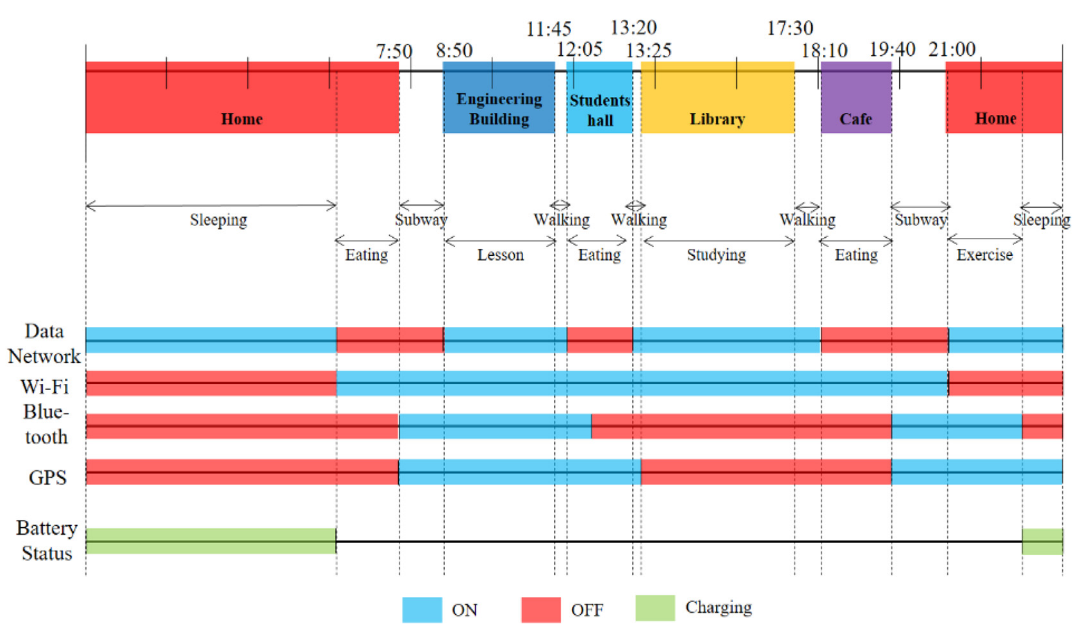

Fig. 6. Situation without device management application.

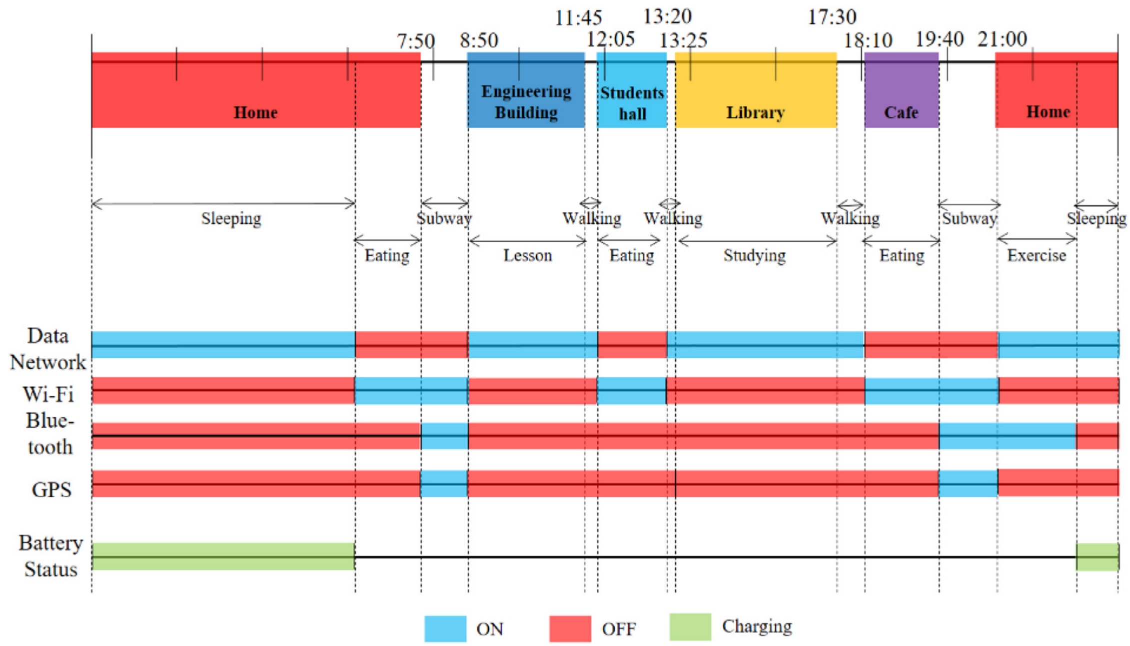

Fig. 7. Situation with device management application.

Table 6.
Accuracy of classifying the state of movement.

Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
Staying	0.947	0.014	0.957	0.947	0.952	0.984
Walking	0.908	0.079	0.912	0.908	0.910	0.960
Running	0.848	0.042	0.828	0.848	0.838	0.966
(Average)	0.901	0.045	0.899	0.901	0.900	0.970

Table 7.
Accuracy of classifying the type of transportation.

Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
Train	0.873	0.052	0.885	0.873	0.879	0.957
Car	0.928	0.072	0.841	0.928	0.883	0.960
Subway	0.762	0.051	0.863	0.762	0.809	0.914
Bus	0.910	0.020	0.833	0.910	0.870	0.969
(Average)	0.860	0.054	0.861	0.860	0.859	0.946

Table 8.
Confusion matrix for movement state.

	a	b	c
a = Staying	1020	43	7
b = Walking	45	1032	46
c = Running	0	69	746

Table 9.
Confusion matrix for type of transportation.

	a	b	c	d
a = Train	1038	47	94	10
b = Car	22	1024	32	25
c = Subway	105	130	860	34
d = Bus	8	16	10	345

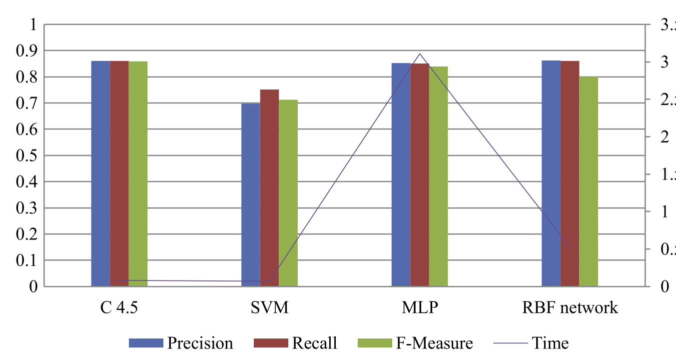

Fig. 8. Comparison of the accuracy and runtime for classifying the type of transportation.

we compare the battery level with and without the proposed system.

4.2. Classification accuracy of movement and transportation

The performance of decision tree to identify the type of transportation is evaluated first. User's movement state can be classified into stop, run and walk. The type of transportation is classified into vehicle, subway, train and taxi. The input was from sensor values of acceleration, magnetic, and orientation. After pre-processing, the accuracy of 10-fold cross validation is used for the input of the variation of axis of sensor, standard deviation and SMA. Tables 6 and 7 show the result of accuracy of the 10-fold cross validation.

Tables 8 and 9 show the confusion matrix of the decision tree for the type of transportation and the state of movement. It can be seen that the state of walking is more discriminative than the other movement states, and subway is highly confused with train or car.

Fig. 8 shows the performance for the type of transportation of several classifiers such as decision tree, SVM, multi-layer perceptron and RBF network. This figure confirms that the decision tree is more stable and faster than other classifiers. To get fairer comparison, we have conducted the *t*-test between decision tree and other classifiers, verifying that all the improvements are statistically significant. The p-values for SVM and RBF network are under .005, and that of multi-layer perceptron is under .001.

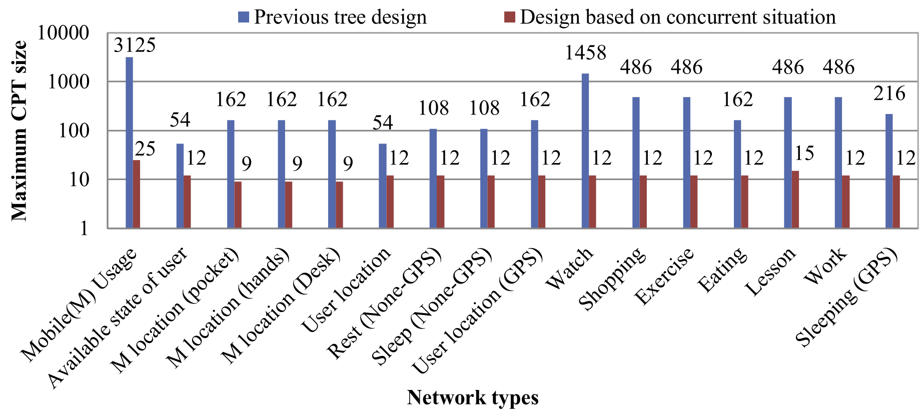

Fig. 9. Change of conditional probability table for each network.

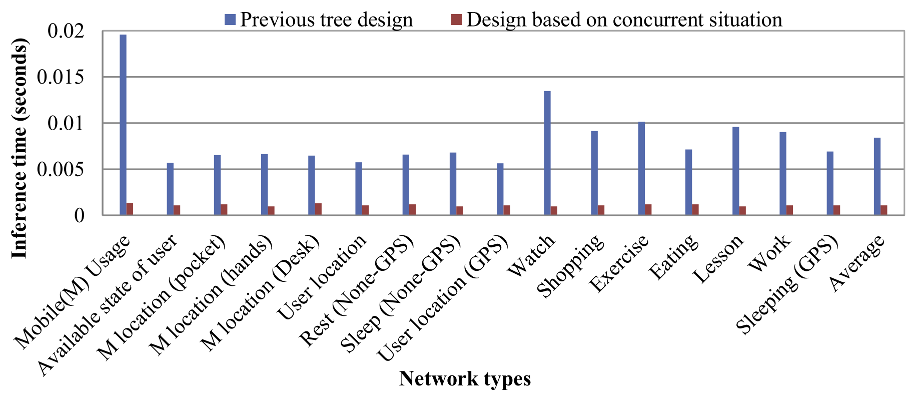

Fig. 10. Change of inference time for each network.

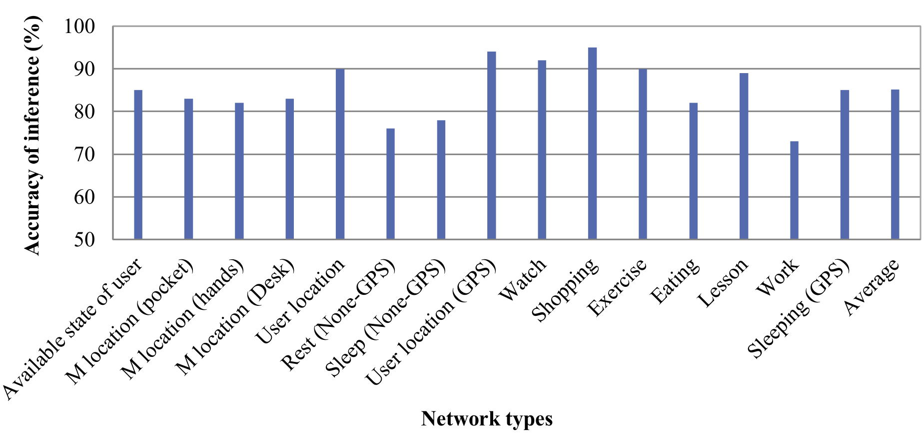

Fig. 11. Accuracy of inference for each network.

4.3. Inference accuracy of situation

In this paper, we designed 16 modular networks to infer the user situation. The parameters of the networks are determined based on the data and related works of national statistics institute of Korea. To reduce the computational overhead, the situation is inferred using linear inference algorithm proposed by Das [18]. The conditional probability table (CPT) is reduced by considering the concurrent situations, so that the computation needed for inference is reduced. Fig. 9 shows the change of CPT for each network.

If we design the parameters based on the previous tree design method, the maximum size of CPT for mobile usage network is 3125. On the contrary, if we design them based on concurrent situation, the size of CPT can be reduced to 25. This causes the CPT

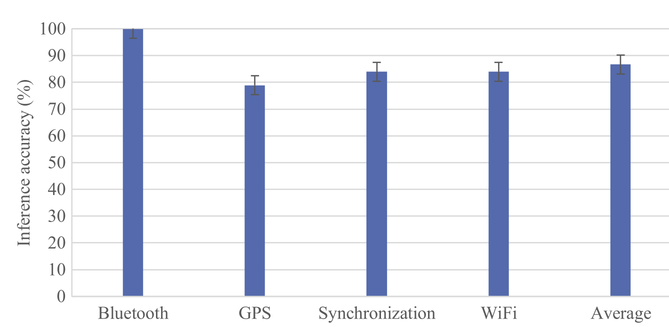

Fig. 12. Accuracy of inferring the superfluous devices.

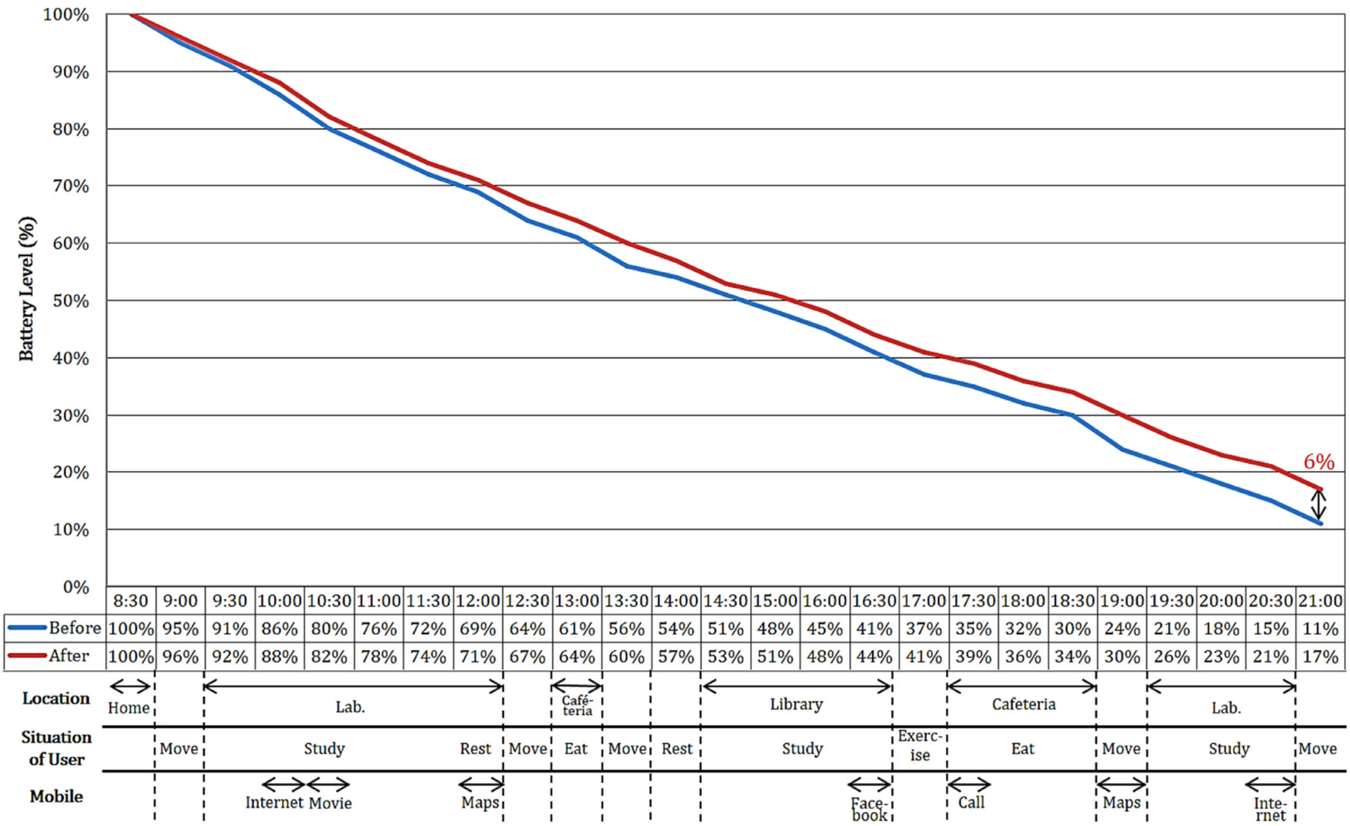

Fig. 13. Comparing battery levels.

size to be proportional to the number of parent nodes and values, resulting in the large reduction of the CPT size in the modular BN.

The time taken for inference was checked to confirm the time reduction. We compared the inference time between tree design and the design based on concurrent situations, and measured the average time after 100 runs of experiments. Fig. 10 shows the result of inference time.

In order to confirm the reduction of computation, we conducted the network experiment. As a result, the size of CPT was reduced from 68% to 92% (Fig. 9). The time required for inference is reduced from 5 times to 13 times (Fig. 10), and the result of accuracy of designed networks is distributed from 76% to 94% (Fig. 11).

4.4. Inference accuracy of unnecessary devices

The other set of modular Bayesian networks is designed to infer idle situation of devices such as Wi-Fi, GPS, data synchronization and Bluetooth. The input values of networks are determined by independent node, and intermediate nodes are designed to reflect the correlation with result node.

We investigate the tendency of users using NEO-PI-R survey [23]. Using the collected data, we measure the accuracy by comparing to the result of inferred networks of unnecessary device. Fig. 12 shows the average accuracy of 85.68% that the proposed system correctly identifies the superfluous devices. The device of the highest accuracy is Bluetooth because of the definite usage pattern of data. On the other hand, the accuracy of GPS device is low because it is used in various situations.

4.5. Comparison on battery levels

In order to evaluate the performance of the proposed device management system in daily lives, we compared the battery usage of one person for a whole day, when he carried together two smartphones all the time. Only one of the devices installed the proposed system. Fig. 13 compares the battery level, resulting in the decrease of battery consumption of about 6% compared to the smartphone without the proposed system. This induces to increase the use time of smartphone for about 2 h per day. Of course, we need to increase the number of persons and the days to get fairer evaluation.

5. Concluding remarks

In this paper, in order to increase the longevity of using smartphone, we proposed a device management system based on hierarchical Bayesian networks of inferring user's situations and unnecessary devices. We designed 16 Bayesian networks to infer user's situations, and four Bayesian networks to infer unnecessary devices of GPS, Wi-Fi, Bluetooth, and synchronization. In order to show the usefulness of the proposed system, we developed a smartphone application for device management. In the experiments with real log data, we confirm that the proposed system results in the accuracy of 85.68% and the reduction of battery consumption of about 6%.

Even though we were successful to show the feasibility of the proposed system, we need several further works to improve the current research. We can divide the future works into short-term and long-term research. In the short-term, we need to incorporate several wearable devices such as smart watch and glasses for total battery saving and develop appealing personalized services with them. In the long-term, we have to improve the context recognition model with deep learning with larger data. We are aiming to develop market sensational applications with the full power of battery saving functionality.

Acknowledgments

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract (UD160066BD).

References

[1] Qualcomm, Designing mobile devices for low power and thermal efficiency, Qualcomm Technologies Report, pp. 1–13, 2013.
[2] Qualcomm, BatteryGuru. https://play.google.com/, 2014.
[3] K.-M. Yang, S.-B. Cho, A low-power consumption context-aware system using modular Bayesian network with linear inference algorithm, in: Proceedings of the 41st KCC Conference, 2014, pp. 1370–1372.
[4] National Statistical Office, Life style research, http://kostat.go.kr/, 2010.
[5] R.R. McCrae, O.P. John, An introduction to the five-factor model and its applications, J. Personality 60 (2) (1992) 175–215.
[6] M. Susi, V. Renaudin, G. Lachapelle, Motion mode recognition and step detection algorithms for mobile phone users, Sensors 13 (2) (2013) 1539–1562.
[7] C. Bettini, et al., Anonymity in location-based services: Towards a general framework, in: International Conference on Mobile Data Management, 2007, pp. 69–76.
[8] M. Weiss, et al., Identifying user traits by mining smart phone accelerometer data, in: International Workshop on Knowledge Discovery from Sensor Data, 2011, pp. 61–69.
[9] E. Munguia, et al., Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor, in: 11th IEEE International Symposium on Wearable Computers, 2007, pp. 37–40.
[10] Z. Zhuang, K.-H. Kim, J.P. Singh, Improving energy efficiency of location sensing on smartphones, in: International Conference on Mobile Systems, Applications, and Services, 2010, pp. 315–330.
[11] J. Parkka, et al., Activity classification using realistic data from wearable sensors, IEEE Trans. Inf. Technol. Biomed. 10 (1) (2006) 119–128.
[12] R. Herrmann, Z. Piero, T. Rosing, Context aware power management of mobile systems for sensing applications, in: Conference on Information Processing in Sensor Networks, 2012.
[13] M. Mohammad, et al., Context-aware mobile power management using fuzzy inference as a service, Mobile Comput. Appl. Serv. (2013) 314–327.
[14] T. Yashiro, et al., A framework for context-aware power management on embedded devices, in: IEEE 1st Global Conference on Consumer Electronics, 2012, pp. 588–592.
[15] X. Zhao, et al., A system context-aware approach for battery lifetime prediction in smart phones, in: ACM Symposium on Applied Computing, 2011, pp. 641–646.
[16] K. Nishihara, I. Kazuhisa, J. Sakai, Power saving in mobile devices using context-aware resource control, in: International Conference on Networking and Computing, 2010, pp. 220–226.
[17] R. Nishkam, et al., Context-aware battery management for mobile phones, in: IEEE International Conference on Pervasive Computing and Communications, 2008, pp. 224–233.
[18] B. Das, Generating conditional probabilities for Bayesian networks: easing the knowledge acquisition problem, arXiv preprint cs/0411034, 2004.
[19] S.-H. Lee, K.-M. Yang, S.-B. Cho, Integrated modular Bayesian networks with selective inference for context-aware decision making, Neurocomputing 163 (2015) 38–46.
[20] F.V. Jensen, Bayesian Networks and Decision Graphs, second ed., Springer, 2007.
[21] K.-S. Hwang, S.-B. Cho, Landmark detection from mobile life log using a modular Bayesian network model, Expert Syst. Appl. 36 (10) (2009) 12065–12076.
[22] C. Xu, V. Srinivasan, J. Yang, Y. Hirase, E.M. Tapia, Y. Zhang, Boe: context-aware global power management for mobile devices balancing battery outage and user experience, in: Proceedings of IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, 2014, pp. 300–308.
[23] P.T. Costa, R.R. McCrae, Age differences in personality structure: A cluster analytic approach, J. Gerontol. 31 (5) (1976) 564–570.

Sung-Bae Cho received the B.S. degree in computer science from Yonsei University, Seoul, Korea and the M.S. and Ph.D. degrees in computer science from KAIST (Korea Advanced Institute of Science and Technology), Taejeon, Korea. He was an Invited Researcher of Human Information Processing Research Laboratories at ATR (Advanced Telecommunications Research) Institute, Kyoto, Japan from 1993 to 1995, and a Visiting Scholar at University of New South Wales, Canberra, Australia in 1998. He was also a Visiting Professor at University of British Columbia, Vancouver, Canada from 2005 to 2006. Since 1995, he has been a Professor in the Department of Computer Science, Yonsei University. His research interests include neural networks, pattern recognition, intelligent man–machine interfaces, evolutionary computation, and artificial life. He was awarded outstanding paper prizes from the IEEE Korea Section in 1989 and 1992, and another one from the Korea Information Science Society in 1990. He was also the recipient of the Richard E. Merwin prize from the IEEE Computer Society in 1993. He was listed in Who's Who in Pattern Recognition from the International Association for Pattern Recognition in 1994, and received the best paper awards at International Conference on Soft Computing in 1996 and 1998. Also, he received the best paper award at World Automation Congress in 1998, and listed in Marquis Who's Who in Science and Engineering in 2000 and in Marquis Who's Who in the World in 2001. He is a Senior Member of IEEE and a Member of the Korea Information Science Society, INNS, the IEEE Computational Intelligence Society, and the IEEE Systems, Man, and Cybernetics Society.

Jae-Min Yu is an M.S. candidate in computer science at Yonsei University. His research interests include evolutionary algorithms and intelligent agents.