A Bayesian Tree Learning Method for Low-Power Context-Aware System in Smartphone

Kyon-Mo Yang, Sung-Bae Cho
Dept. of Computer Science
Yonsei University
Seoul, Korea
kmyang@sclab.yonsei.ac.kr, sbcho@yonsei.ac.kr

***Abstract*—Context-aware services using smartphone have been proliferated for ubiquitous computing. However, the capacity of smartphone battery is extremely limited so that the services cannot be effectively used. In this paper, we propose a low-power context-aware system using tree-structured Bayesian network. Bayesian network, one of the probabilistic models, is known to handle the uncertainty flexibly. A well-known problem of the probabilistic model, however, is high time complexity, which leads to significant consumption. To reduce the time complexity, we propose a tree-structure learning method. The key idea lies in how to consider the relation of each node. For the reason, we conduct the spanning tree based on the mutual information among nodes. The data for experiment were collected from Android phone for two weeks. The amount of the collected data is 7,464. The accuracy of proposed method achieves 94.13%. The energy consumption is measured using the power tutor application.**

Keywords-Low-power consumption, context-awareness, tree-structure Bayesian network, Structure learning.

I. INTRODUCTION

Recent proliferation of the smartphones leads to developing a large variety of applications and investigating on the use of various sensors through context-awareness. Previous researchers focus on the accuracy of context-awareness using all possible sensors [1]. The battery capacity of smartphone is well behind the development of service application. In a typical case, the user has to carry an extra battery or charge it frequently. There is the critical issue of how to reduce the battery consumption for the context-awareness in smartphone.

In this paper, we propose a low-power context-aware system using tree-structure Bayesian network. Bayesian network is one of the powerful probabilistic methods for context-awareness [2]. It can infer context in uncertain situation or with the incomplete data. However, the probability model generally has high time complexity, because the model has to calculate the probability of each node every time. It causes the significant consumption for context-awareness in smartphone. We propose a tree-structure learning method to reduce the time complexity.

We compare the accuracy using different structure learning methods and evaluate the time complexity of the proposed method. In addition, we verify the low-consumption feature of the proposed method in a real smartphone environment.

The paper is organized as follows. Section 2 presents the related works for context-awareness, battery problem, and Bayesian networks. Section 3 describes in detail the proposed low-power context-aware system. Finally, section 4 reports the experiments conducted to compare the power consumption of the proposed method and the monolithic BN.

II. RELATED WORKS

A. *Context-aware service in smartphone*

Context-aware services aim to provide the convenient services to users who are in the contexts recognized. Context is all the information related to the interactions between user and applications [3]. Interactions are becoming important as research in pervasive computing progresses. Context-aware services in smartphone recognize the situation and provide services. The applications are developed using various sensors.

Lee and Cho proposed a method using the KeyGraph and Bayesian network to infer mobile life log [4]. Ravi, et al. proposed battery management service [5]. The service recommends the battery charging time depending on current state. Phithakkitnukoon and Dantu proposed a three-step approach in designing the model based on the embedded sensor data for controlling alert mode [6]. Lester, et al. presented the approach to building a system that exhibits these properties and provided evidence based on data for 8 different activities; Sitting, standing, walking, walking up stairs, walking down stairs, riding elevator down, riding elevator up, and brushing teeth [7]. Santos, et al. described the architecture, operation and potential applications of a prototype system developed within the User-Programmable Context-aware Services (UPCASE) project [8]. A lot of applications have applied to the context-awareness. However, these previous works lack of dealing with the power consumption of smartphone.

B. *Battery lifetime problem*

Fig. 1 shows the key features of mobile devices. According to the TIME Mobility Poll, conducted in June and July 2012, the main issue for mobile users is battery life. 62 percent of American mobile users wish for improvements in that area. Device size does not seem to be a problem for most of them, as only 5 percent of American users want a smaller device. Because the battery life is short, the user has to carry an extra battery or charge it frequently. There is the critical issue of how to reduce power consumption for context-awareness.

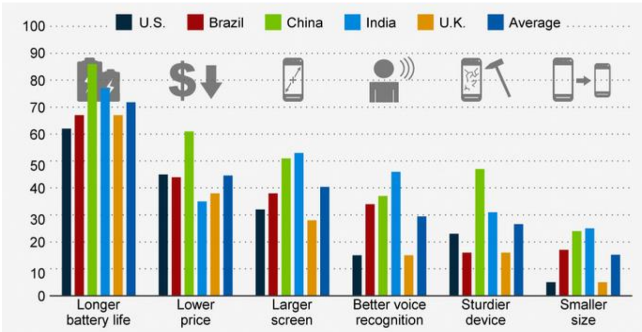


Figure 1. Key features of mobile devices by TIME Mobility Poll.

Many researchers have proposed low-power application using context-awareness for solving the problems. Seo, et al. proposed a context-aware configuration manager for smartphones [9]. The system changes the configuration of a smartphone according to the user-defined policy rules. Bareth and Kupper proposed a hierarchical positioning algorithm [10]. The algorithm dynamically deactivates different positioning technologies and only activates the positioning method with the least energy consumption. Miraoui, et al. proposed limited resources-aware service [11]. The system changes the services considering the current resources of mobile phone. These researchers lack of reducing the power consumption of the context-aware module. In this paper, the learning method maintains the advantage of probabilistic model and reduces the time complexity for the power consumption.

C. Bayesian networks

Bayesian network is devised as a powerful technique for handling the uncertainty. Bayesian network has a structure of a directed acyclic graph which represents the link relations of the node, and has conditional probability tables (CPT). Assume that a node is independent of each other.

$$\begin{aligned} P(U) &= P(A_1, A_2, \ldots, A_n) \\ &= P(A_1)P(A_2 \mid A_1)\ldots P(A_n \mid A_1, A_2, \ldots, A_{n-1}) \\ &= \prod_{i=1}^{n} P(A_i \mid pa(A_i)). \end{aligned} \qquad (1)$$

The conditional probability distribution of variable A can be represented as $P(A \mid pa(A))$, where $pa(A)$ denotes the set of parent variables of variable A, where U is a set of node, and the joint probability distribution is computed by the chain rules as (1).

There are two approaches to identify the structure and parameter of Bayesian network. The first approach is to learn model from the data on problem domains. The structure learning is useful if we have a lack of understanding about the system. The method requires the sufficient amount of data, but it is not easy to obtain reliable data in many real-world problems. The second one can be construct it based on the domain knowledge. The experts identify the structure and set of parameters according to their knowledge, if we do not have enough data in the domain.

The time complexity of the Bayesian network is calculated using the LS algorithm as follows [12]. Here, n represents the number of nodes, k represents the maximum number of parents for each node, r denotes the number of values for each node, and w represents the maximum number of clique.

$$CMPX = O(k^3 n^k + wn^2 + (wr^w + r^w)n) \qquad (2)$$

D. Structure design of Bayesian network

TABLE I. RELATED WORKS FOR REDUCING INFERENCE TIME

Authors	Necessity of knowledge	Extra procedure	Description
Pearl [13]	X	O	Using only relevant CPT about current inference
Heckerman and Breese [14]	O	X	Removing the uncertain interaction between causes and effects
Zhang and Poole [15]	X	X	Removing weak dependencies before inference
Kjaerulff [16]	X	O	Removing weak dependencies before inference
Koller and Pfeffer [17]	O	X	Applying object concept to BN
Oude, et al. [18]	O	X	Hierarchical modular approach designed for multiple agents

Bayesian network is a robust tool for practical problems which involve high level of uncertainty. However, utilizing it in the large-scale domains is difficult because considerable effort is put on designing and maintaining the network. Besides, it is unable to entirely apply on ubiquitous devices since lots of computation power and resources are required in the inference process. For these reasons, there have been many studies on reducing the time complexity. Table I shows two types of the related works for reducing inference time. First, the necessity of knowledge implies that the network is not designed automatically. Second, when the system infers, it needs extra procedure for modifying the network structure.

The Noisy-OR model was proposed by Pearl [13]. The model can compute the distributions required for the CPT from a set of distributions, elicited from the expert, and the magnitude which grows linearly with the number of parents. Heckerman and Breese proposed an extended version of the method called Noisy-MAX Gate [14]. This method showed a collection of conditional independence assertions and functional relationships and removed the representation of

 ISBN: 978-1-61208-353-7

the uncertain interactions between cause and effect. Zhang et al. removed weak dependencies before inference [15]. The method evaluated the relation of each node with query node and modified the structure of network through removing the nodes and edges. Kjaerulff presented a method for reducing the computational complexity through removal of weak dependences [16]. Koller and Pfeffer proposed a method where Bayesian network is applied to an object concept called OOBN [17]. This method used a Bayesian network fragment to describe the probabilistic relations between the attributes of an object. Oude divided BN model into several smaller multi-level modules and inferred each module sequentially from the low level to the high level [18]. Its composition is similar to MBN, but it restricts the networks in hierarchical structure.

The previous works in the table have problems. If the design method needs domain knowledge, it requires the time to analyze the domain. If the system needs extra procedure, it consumes extra battery. To solve these problems we propose a tree structure learning method.

III. LOW-POWER CONTEXT-AWARE SYSTEM

In this paper, we propose the low-power context-aware system. The proposed method considers for the power consumption of inference module.

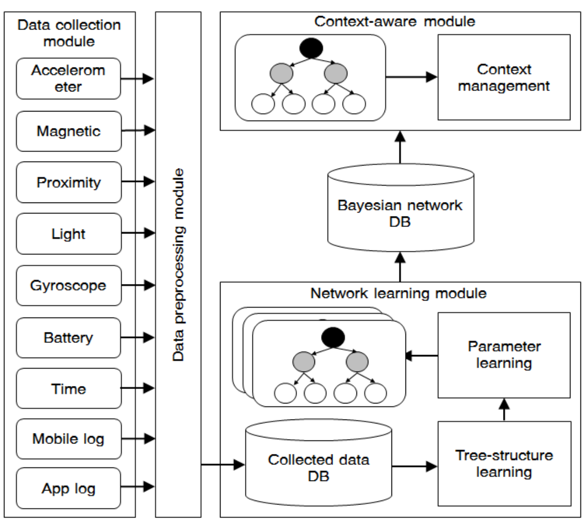


Figure 2. System architecture

Fig. 2 illustrates the system architecture. The system consists of four modules: Sensor collection, data preprocessing, network design, and context-awareness. In this system, we do not use the sensors that require high power consumption. The sensor collection module obtains the continuous sensor data in smartphone. The data are sent to the data preprocessing module that discretizes them using decision tree. The network learning module trains the structure and parameter of the BN. The context-awareness module infers user situation using the tree-structure Bayesian network. If the result of inference is higher than the threshold, it is the current situation.

A. *Sensor data preprocessing*

The proposed system focuses on how to reduce the power consumption for context-awareness using sensor information. Therefore, the other source that generates the energy consumption such as memory, synchronization, and so on, is not considered in the system. Abdesslem, et al. measured the energy consumption of different sensors as shown in Table II [19]. Each sensor runs continuously on a Nokia N95 8GB smartphone until the battery was depleted. In this research, the power consumption of GPS is 623mW, and it is 6 times more power consumption than the accelerometer sensor.

TABLE II. POWER CONSUMPTION IN SENSORS

Sensors	Battery life (hrs)	Average power consumption (mW)
Camera	3.5	1258
IEEE 802.11	6.7	661
GPS	7.6	623
Microphone	13.6	329
Bluetooth	21.6	211
Accelerometer	45.9	96

We select the sensors which can use during half-day because of the battery life time.

X_axis	Y_axis	Z_axis	Orientatic	Pitch	Roll	Magnetic1	Magnetic2	Magnetic3	Proximity	light	Gyro1	Gyro2	Gyro3
-0.22984	8.389283	8.283937	78.09244	-57.7119	-6.56882	-37.38	-18.48	-28.5	8	4435	0.060781	-0.67562	-0.12126
2.27928	7.967903	5.477933	82.18597	-49.7656	-3.5679	-41.7	-13.5	-24.12	8	7461	0.376598	3.172834	0.534202
0.191536	5.48751	5.822699	63.09831	-43.964	-8.35287	-31.2	-0.36	-42.12	8	3754	0.460898	2.666427	-0.57879
-3.00712	5.315128	11.21444	71.23709	-18.1507	-19.5634	-17.34	1.44	-44.88	8	4353	-2.03082	-1.85245	-1.62215
1.436521	3.533842	5.554548	55.31022	-30.242	-6.72346	-37.56	-2.7	-35.58	8	10003	-0.91935	0.572381	0.939511
-0.16281	-3.75411	9.844957	63.49014	8.344509	3.910431	-41.76	30.24	-25.8	8	11076	1.686599	0.640798	0.586125
1.647211	7.31668	-1.78129	18.79346	-2.84995	3.336133	-30.6	31.68	-32.76	0	121	-2.83105	-2.25104	1.138653
-3.5913	2.911349	-2.62405	250.4009	-31.2422	-54.0916	27.66	25.62	7.74	0	304	-2.16613	1.440115	1.304503
-7.67102	10.92714	-1.94409	244.0992	-52.0354	-36.8056	35.4	1.56	24.48	0	147	1.385442	-4.84844	1.579697
-2.78685	13.81933	-6.6463	284.0452	-74.0556	-9.47077	41.22	-16.5	-7.56	0	8	1.054048	-1.51098	1.498758
-5.77481	9.433155	-3.47638	272.1701	-68.8965	-21.1027	37.8	-22.08	8.22	0	80	-3.43551	-2.70033	-1.18752
-8.19775	4.520253	3.926491	267.1496	-77.119	-12.7805	40.44	-14.64	-11.64	0	4	1.169502	5.180137	-0.91996
-13.3692	9.356541	0.938527	291.223	-56.512	-26.1563	37.08	-15.06	10.68	0	31	-2.49813	-6.2409	-0.98471
-7.76679	8.465898	-2.16436	248.3102	-66.6607	-23.3265	42.36	-4.86	-7.8	0	2	1.469436	1.490206	-0.40897
-14.1545	13.54161	-3.07416	258.3507	-45.9125	-44.0226	36.9	-6.42	14.58	0	12	-1.87749	-3.80203	-1.88299
-7.87214	5.181053	-1.10133	284.9868	-57.2688	-23.4543	42.6	4.5	-5.04	0	3	-2.37657	-1.95752	2.866791
-6.42604	11.98059	-1.87705	256.9636	-38.9338	-49.3785	37.38	-1.02	18.36	0	4	-0.99632	0.311541	-0.74037

Figure 3. An example of the sensor data collected

Fig. 3 shows an example of the sensor data collected. If these continuous data were used as the states of the input values, the time complexity of network would be very high. It affects to increase the size of CPT because it is related to the number of the states of values [2]. In other word, if the continuous values map to the states of the value, the number of state is almost infinite. For this reason, the states of sensor data are preprocessed.

TABLE III. DEFINITION OF INPUT AND OUTPUT

Type	Sensors	Values
Input	Sensor: Accelerometer: X_axis	{Low, Middle, High}
	Sensor: Accelerometer: Y_axis	{Low, Middle, High}
	Sensor: Accelerometer: Z_axis	{Low, Middle, High}
	Sensor: Accelerometer: Orientation	{Low, Middle, High}
	Sensor: Accelerometer: Pitch	{Low, Middle, High}
	Sensor: Accelerometer: Roll	{Low, Middle, High}
	Sensor: Magnetic:1	{Low, Middle, High}
	Sensor: Magnetic:2	{Low, Middle, High}

	Sensor: Magnetic:3	{Low, Middle, High}
	Sensor: Proximity	{Low, Middle, High}
	Sensor: Light	{Low, Middle, High}
	Sensor: Gyro:1	{Low, Middle, High}
	Sensor: Gyro:2	{Low, Middle, High}
	Sensor: Gyro:3	{Low, Middle, High}
Output	User: Situation	{Sleeping, Exercising, Moving street, Having meal, Shopping, Studying, Viewing}

There are two discretization techniques. First, the range of input values can be divided into a predefined number of intervals of equal width. Second, it can be divided using statistical methods. A decision tree is one of the powerful and popular tools for making rules. All the continuous input such as accelerometer, gyroscope, and so on make the rules with a range of the division using the decision tree, because the input data do not need to change into the semantic data. It just needs to divide three ranges: Low, middle, and high.

The General Social Survey (GSS) collected the data on social trends in order to monitor changes in the living conditions [20]. The survey defined the category of situation. The output of network is defined with referring to the survey. Table III represents the input values and output values.

B. *Tree-structure learning*

The purpose of tree-structure learning is to reduce the time complexity by considering the relation of each node. The proposed method does not need the extra computational time for modifying the structure. In addition, the method does not need expert's knowledge because of learning from the collected data. Fig. 4 shows the flowchart of the proposed method. The learning method consists of six steps.

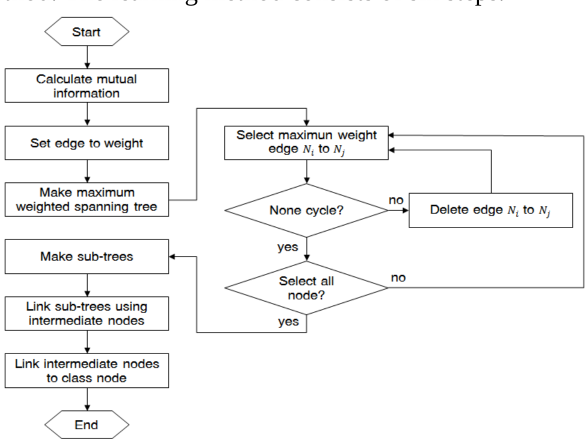


Figure 4. Flowchart of tree-structure learning method

First, we calculate mutual information with the relation of class. The mutual information is to calculate the relation between the attribute *X* and attribute *Y* from data as (3).

$$I_{ij}(N_i;N_j/C)=\sum_{n_i\in N_i,n_j\in N_j,c\in C}P(n_i,n_j,c)\log\frac{P(n_i,n_j/c)}{p(n_i/c)p(n_j/c)} \quad (3)$$

The attributes N_i and N_j are used for input nodes in the network. The class *C* represents the output node in the network.

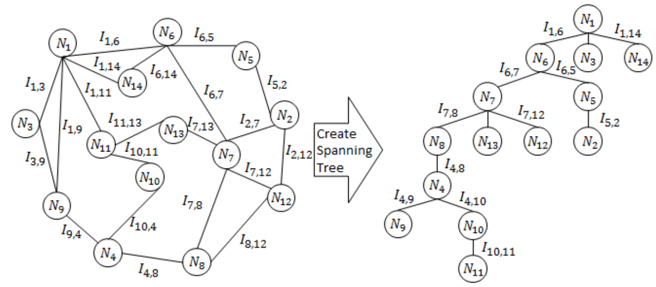


Figure 5. Creation result of maximum spanning tree

Next, maximum spanning tree is created using the calculated mutual information. The mutual information sets the weight of spanning tree. The tree randomly selects one node. Then, it selects another node that has maximum weight from the node. If the link does not create cycle, the node is selected. This procedure is repeated until all nodes are selected. Fig. 5 shows this step. In this figure, node 1 is root node. That is selected randomly. Nodes 6, 3, and 14 have the higher degree of association to the node 1 than other nodes. Nodes 7 and 5 have the highest degree of association to node 6.

Input:
Node *N*, Mutual information I_{ij}
Output:
Maximum spanning tree

Setting the link weight to I_{ij} ;

Select node N_i ;
While (Not all nodes *N* are selected)
Link N_i to N_j , I_{ij} has the maximum weight;
If (Cycle is created by the link)
Delete link N_i to N_j ;
Return to the first step;

Figure 6. Creation algorithm of maximum spanning tree

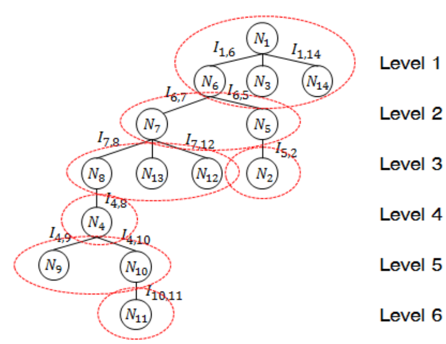


Figure 7. Grouping the maximum spanning tree

 ISBN: 978-1-61208-353-7

Fig. 6 shows the creation algorithm of maximum spanning tree. The network can maintain the relation of each node though this algorithm. Third, the method constructs a sub-tree through grouping some nodes by considering the relation of each node. Fig. 7 shows how to make sub-trees when the group level is 1. The group level means how many depths can be grouped.

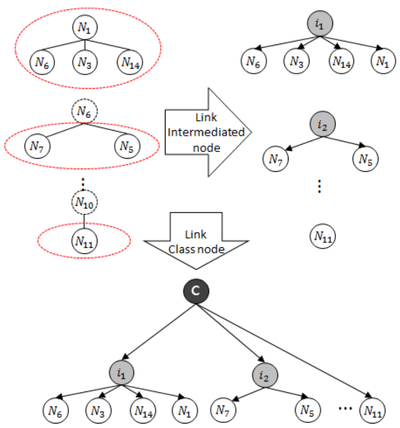


Figure 8. Creating tree-structured network

The group links to an intermediated node, which is used for considering relation of the grouped nodes. Finally, the class node links all intermediate nodes. Fig. 8 shows the final step. The parameters of the network are trained using Maximum Likelihood Estimation (MLE) [2].

IV. EXPERIMENTAL RESULTS

A. Experimental setting

The data were collected from three graduate students for two weeks. We used the Samsung Galaxy S3. Android phone collects sensor data twice per a second, and the amount of the collected data is 7,464.

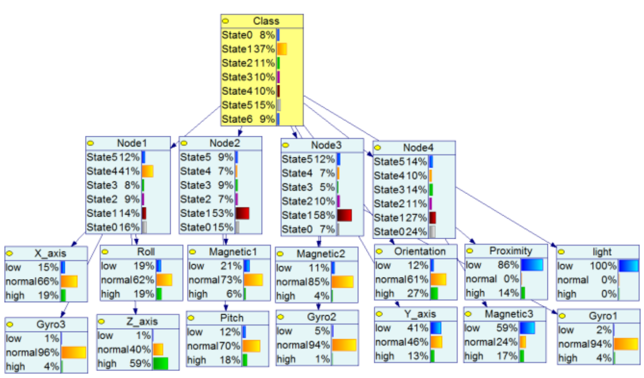


Figure 9. Learned tree-structured network

We collected on seven situations: Sleeping, exercising, moving street, having meal, shopping, studying, and viewing. When they collected the data, the smartphone was put into their pocket. The students selected the situation and conducted it. We learned the network using seven sensor data: Accelerometer, magnetic, gyroscope, light, orientation, pitch, and roll. Fig. 9 shows the tree-structured network using the proposed structure learning method. The network consists of fourteen input nodes, four intermediate nodes, and one output node. The monolithic BN is also trained using EM algorithm [2].

B. Time complextity

This experiment verifies that the proposed method has lower time complexity than other methods. We calculate it using LS algorithm. We compare monolithic BN (BN), Tree-agumented BN (TAN) and the proposed method. It is assumed that the number of clique w equals numbers of parents k . The maximum number of states is 7. The maximum number of parent node of monolithic BN is 7 while the maximum number of parent node of the proposed BN is 1. The number of state of the intermediate module changes 2 to 15. Fig. 10 shows the time complexity of each method.

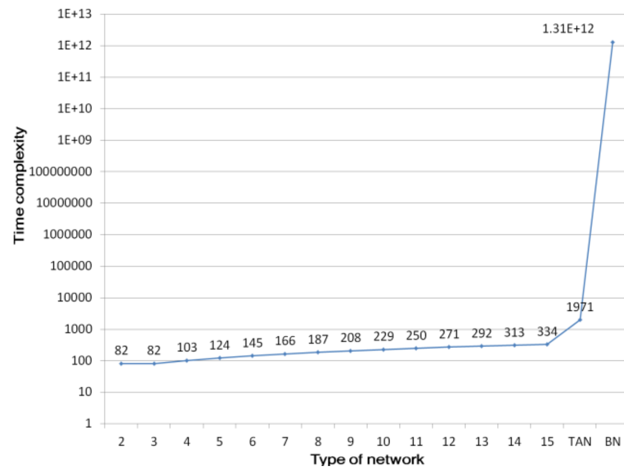
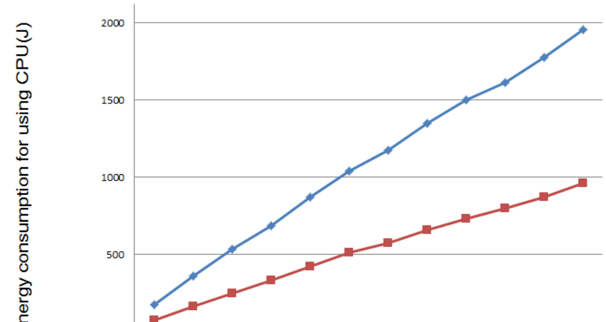


Figure 10. Time complexity usng EM algorithm



	1	2	3	4	5	6	7	8	9	10	11	12
Conventional BN	175	364	536	684	874	1038	1175	1345	1498	1612	1776	1954
Proposed BN	74.7	163	252	335	422	510	577	657	730	797	873	960

Figure 11. Comparison of energy consumption

The experimental result shows that the time complexity is slightly increased according to the number of states in the proposed network. However, the time complexity of the monolithic BN is dramatically increased in comparison with the proposed network. To verify the relation between

 ISBN: 978-1-61208-353-7

reducing the time complexity and reducing the energy consumption, we measured the energy consumption using power tutor application [21]. The application infers 100 times per a second. Fig. 11 confirms the difference of the time consumption of the monolithic BN (BN) and the proposed BN (PBN). Although the monolithic BN consumes the 1,954*J* for an hour, the proposed BN consumes the 960*J* for an hour.

C. *Comparison of accuracy*

We conduct 10-fold cross validation to calculate the accuracy of each network as shown in Fig. 12.

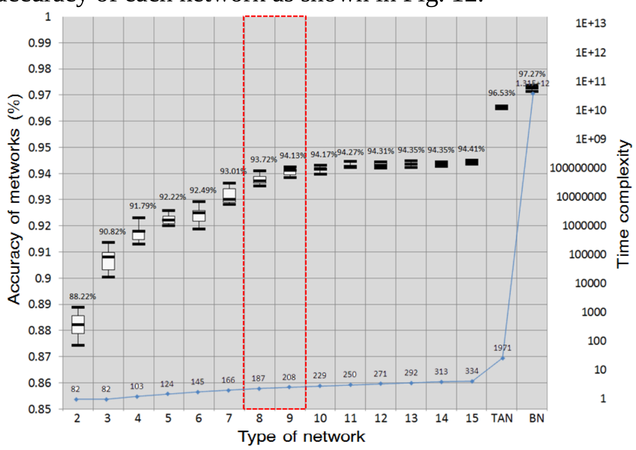


Figure 12. Accuracy of the networks

As a result, although the proposed method has slightly lower accuracy than monolithic BN, there is the relatively small difference of the time complexity. If the system selects eight or nine as the number of states, the accuracy of the network is 93.72%, although the system can have better battery life than monolithic BN.

V. CONCLUDING REMARKS

In this paper, we have proposed tree-structure learning method for a low-power context-awareness. The method does not require the extra computational time and the domain knowledge considering the relation of each node. The system is aware of seven situations. To verify the efficiency of the proposed system, we compare the accuracy of the proposed method against the monolithic Bayesian network and calculate the time complexity. In addition, we confirm the power consumption using power tutor application and verify that the system has lower consumption than the monolithic BN. We will improve the method through modular approach by considering the relation of nodes. The system will be applied to various context-aware service applications.

ACKNOWLEDGMENT

This work was supported by Samsung Electronics, Inc.

REFERENCES

[1] G. Chen, D. Kotz, "A survey of context-aware mobile computing research," Technical Report, Dept. of Computer Science, Dartmouth College, vol. 1, no. 2, pp. 1–16, 2000.

[2] F. V. Jensen, Bayesian Networks and Decision Graphs," Springer, 2007.

[3] A. K. Dey, "Understanding and using context," Personal and Ubiquitous Computing, vol. 5, no. 1, pp. 4–7, 2001.

[4] Y. S. Lee and S.-B. Cho, "Extracting meaningful contexts from mobile life log," Intelligent Data Engineering and Automated Learning, vol. 4881, pp. 750–759, 2007.

[5] N. Ravi, J. Scott, L. Han, and L. Iftode, "Context-aware battery management for mobile phones," Pervasive Computing and Communications, pp. 224–233, March, 2008.

[6] S. Phithakkitnukoon and R. Dantu, "Context-aware alert mode for a mobile phone," IEEE Pervasive Computing and Communications, vol. 6, no. 3, pp. 1–23, 2010.

[7] J. Lester, C. Tanzeem, and G. Borriello, "A practical approach to recognizing physical activities," Pervasive Computing, vol. 3968, pp. 1–16, 2006.

[8] A. C. Santos, G. M. P. Cardos, D. R. Ferreira, P. C. Diniz, and P. Chainho, "Providing use context for mobile and social networking applications," Pervasive and Mobile Computing, vol. 6, no, 3, pp. 324–341, 2010.

[9] S.-S. Seo, A. Kwon, J. M. Kang, J. Strassner, and J. W. Hong, "PYP: Design and implementation of a context-aware configuration manager for smartphones," Proc. of Int. Workshop on Smart Mobile Applications, pp.12–15, June, 2011.

[10] U. Bareth and A. Kupper, "Energy-efficient position tracking in proactive location-based services for smartphone environments," Computer Software and Applications, pp. 516–521, July, 2011.

[11] M. Miraoui, C. Tadj, J. Fattahi and C. B. Amar, "Dynamic context-aware and limited resources-aware service adaptation for pervasive computing," Software Engineering, vol. 2011, pp. 1-11, July, 2011.

[12] V. K. Namasivayam and V. L. Prasanna, "Scalable parallel implementation of exact inference in Bayesian networks," Conf. Parallel and Distributed Systems, vol. 1, pp. 8–16, July, 2006.

[13] J. Pearl, Bayesian Networks, University of California, Los Angeles, CA 90095, 2011.

[14] D. Heckerman and J. S. Breese, "Causal independence for probability assessment and inference using Bayesian networks," IEEE Trans. on Systems, Man and Cybernetics, vol. 26, no. 6, pp. 826–831, 1996.

[15] N. L. Zhang and D. Poole, "A simple approach to Bayesian network computations," Proc. of Conf. on Artificial Intelligence, pp. 171-178, May, 1994.

[16] U. Kjaerulff, "Reduction of computational complexity in Bayesian networks through removal of weak depedencies," Proc. of Conf. on Uncertainty in Artificial Intelligence, pp. 374-382, 1994.

[17] P. Weber, and L. Jouffe, "Complex system reliability modelling with dynamic object oriented Bayesian networks (DOOBN)," Reliability Engineering and System Safety, vol. 91, no. 2, pp. 149–162, 2006.

[18] P, De Oude, G. Pavlin, and T. Hood, "A modular approach to adaptive Bayesian information fusion," Information Fusion, pp. 1–8, 2007.

[19] F. B. Abdesslem, A. Phillips, and T. Henderson, "Less is more: Energy-efficient mobile sensing with senseless," Proc. of Workshop on Networking, Systems, and Applications for Mobile Handhelds, pp. 61–62, 2009.

[20] "General Social Survey," http://www23.statcan.gc.ca/, 2010.

[21] "A power monitor for android-based mobile platforms," Available from: http://powertutor.or