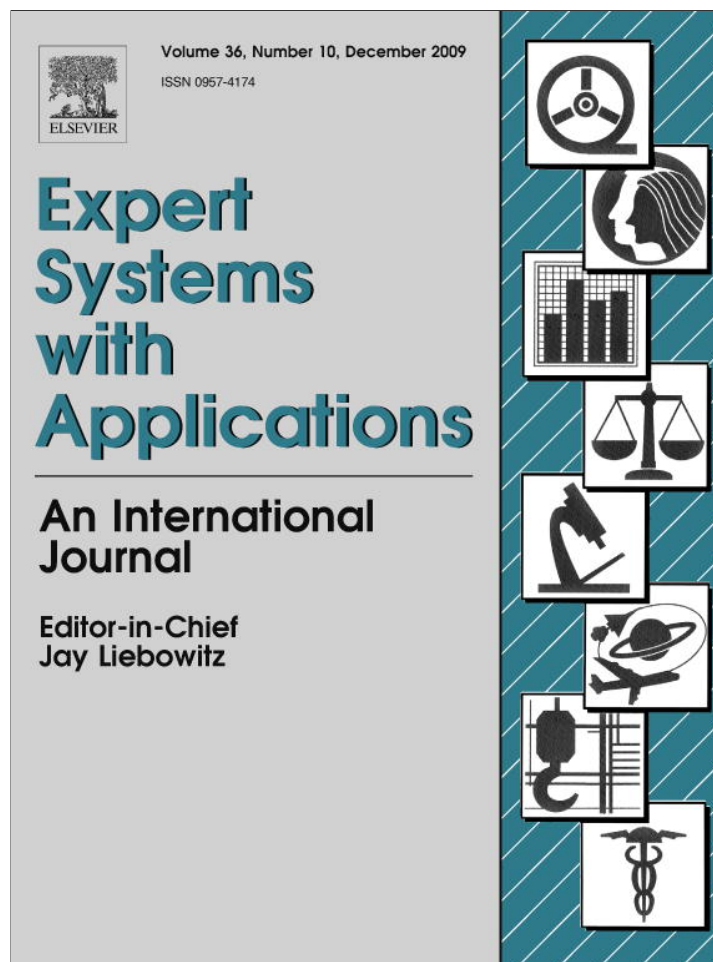


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Expert Systems with Applications

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

Landmark detection from mobile life log using a modular Bayesian network model

Keum-Sung Hwang, Sung-Bae Cho *

Department of Computer Science, Yonsei University, 262 Seongsanro, Sudaemoon-ku, Seoul 120-749, South Korea

ARTICLE INFO

Keywords:

Modularized probabilistic reasoning
Mobile application

ABSTRACT

Mobile devices can now handle a great deal of information thanks to the convergence of diverse functionalities. Mobile environments have already shown great potential in terms of providing customized services to users because they can record meaningful and private information continually for long periods of time. Until now, most of this information has been generally ignored because of the limitations of mobile devices in terms of power, memory capacity and speed. In this paper, we propose a novel method that efficiently infers landmarks for users to overcome these problems. This method uses an effective probabilistic Bayesian network model for analyzing various kinds of log data in mobile environments, which were modularized in this paper to decrease complexity. We also present a cooperative inference method, and the proposed methods were evaluated with mobile log data generated and collected in the real world.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Mobile environments have very different characteristics from desktop computer environments. First of all, mobile devices can collect and manage various kinds of user information, for example, by logging a user's calls, SMS (short message service), photography, music-playing and GPS (global positioning system) information. Also, mobile devices can be customized to fit any given user's preferences. Furthermore, mobile devices can collect everyday information effectively. Such features allow for the possibility of diverse and convenient services, and have attracted the attention of researchers and developers. Recent research conducted by Nokia is a good example (Nokia LifeBlog). Especially, the context-aware technique that has recently been widely studied is more applicable to mobile environments, so many intelligent services such as intelligent calling services (Schmidt, Takaluoma, & Mntyjarvi, 2000), messaging services (Lo, Thiemjarus, & Yang, 2003), analysis, collection and management of mobile logs (DeVaul, Sung, Gips, & Pentland, 2003; Gemmell, Williams, Wood, Lueder, & Bell, 2004; Korpipaa, Mantyjarvi, Kela, Keranen, & Malm, 2003; Krause, Smailagic, & Siewiorek, 2006; Raento, Oulasvirta, Petit, & Toivonen, 2005; Siewiorek et al., 2003; Zheng & Ni, 2006) have been actively investigated.

However, mobile devices present some limitations. They contain relatively insufficient memory capacity, lower CPU power (data-processing speed), smaller screen sizes, awkward input interfaces, and limited battery lives when compared to desktop

PCs. In addition, they have to operate in the changeable real world, which means that they require more active and effective adaptation functions (Dourish, 2004).

In this paper, we propose a novel method of analyzing mobile log data effectively and extracting semantic information and memory landmarks, which can be used as special ways of helping recall specific functions (Horvitz, Dumais, & Koch, 2004). The proposed method adopts a Bayesian probabilistic model to efficiently manage various uncertainties that can occur when working with mobile environments, including real-world irregularities, like varying levels of attention and emotions, inaccuracy of sensors, and uncertain causal factors. The proposed model uses a cooperative reasoning method with a modular Bayesian network (BN) in order to work competently in mobile environments. We also discuss how to learn and update the Bayesian inference model by using the proposed BN learning method with training data. The proposed method was applied to several experiments using both synthetic data and real mobile log data collected with a smartphone for a month in the real world. In Hwang and Cho (2006), we proposed a method for identifying landmarks on mobile devices, which was a modularized BN model designed by human manually.

There have already been various attempts to analyze log data and to support expanded services by using the probabilistic approach. Li and Ji used a probabilistic model for active affective state detection of user (Li & Ji, 2005). They utilized a dynamic Bayesian network and the utility theory to reason the 'fatigue,' 'nervous,' and 'confused' states. They showed that the probabilistic approach was good at management of uncertain information like affection.

Ji et al. used also a dynamic Bayesian network method for real-time monitoring human fatigue (Ji, Lan, & Looney, 2006). They

* Corresponding author. Tel.: +82 2 2123 2720; fax: +82 2 365 2579.

E-mail addresses: yellowg@cs.yonsei.ac.kr (K.-S. Hwang), sbcho@cs.yonsei.ac.kr (S.-B. Cho).

concerned and combined many conditions such as light, heat, humidity, time, napping, anxiety, temperature, weather, and work type. Zhang and Ji proposed an active and dynamic information fusion method for multi-sensor systems with dynamic Bayesian networks (Zhang & Ji, 2006). This showed the usefulness of Bayesian approach for information fusion. These works showed the Bayesian probabilistic approach was good tool for handling, reasoning, and combining uncertain information.

Krause et al. clustered the sensor and log data collected on mobile devices, discovered a context classifier that reflected a given user's preferences, and estimated the user's situation in order to provide smart services (Krause et al., 2006). The context classifier was constructed using the BN model, which was based on a general learning method for a small domain of classification subjects. Horvitz et al. proposed a method that detected and estimated landmarks by discovering a given human's cognitive activity model from PC log data based on the Bayesian approach (Horvitz et al., 2004). Their approach showed a good performance for recognizing and learning everyday-PC-life of humans.

However, these methods were not suitable for mobile devices that were limited in terms of capacity and power. For larger domains, the general BN and BN learning method require highly complex computation. This is a crucial problem when it comes to modeling everyday life situations with mobile devices. To overcome these problems, a more appropriate approach was required to reduce the complexity levels. Marengoni, Hanson, Zilberstein, and Riseman (2003) tried to reduce the complexity levels of the BN model by dividing it into several multi-level modules and using procedural reasoning of the connected BNs (just like chain inference). However, this method required procedural and classified properties of the target functions.

Tu, Allanach, Singh, Pattipati, and Willett (2006) proposed a hybrid BN model that allowed hierarchical hybridization of BNs and HMMs. However, it supported only links from lower level HMMs to higher level BNs without consideration of links between BNs of the same level. Hwang, Kim, and Zhang (2006) proposed the hierarchical probabilistic graphical modeling method, which constructed a hierarchical and distributed BN structure using generated hidden nodes and the links between them. However, this method is improper in mobile and private service environments since it leads to an increase of the number of nodes and does not keep the intuitive causal structure.

2. Landmark inference from mobile log data

The overall process of landmark extraction from the mobile log data used in this paper is shown in Fig. 1. Various mobile log data

is preprocessed in advance, and then the landmark-reasoning module detects the landmarks. The preprocessing module is operated by the techniques of pattern recognition and simple rule reasoning. The BN reasoning module performs probabilistic inference. The update module learns the Bayesian network models and adapts to the user and environment by using the accumulated data.

BNs refer to models that can express large probability distributions with relatively small costs to statistical mechanics. They have the structure of a directed acyclic graph (DAG) that represents the link (arc) relations of the node, and has conditional probability tables (CPTs) that are constrained by the DAG structure (Korb & Nicholson, 2003). Fig. 2 shows an example BN that was designed by human and used for the application of this paper. It shows a DAG structure, node name, state name and inferred probabilities.

2.1. Collection and preprocessing

Table 1 shows log information collected on a mobile device and on the internet. The GPS log presents the places that the user visited, and the call and SMS logs show the user's calling patterns. The MP3 (a music file format) player log offers an idea of the user's emotions and the photograph log shows when the user wanted to memorize something.

Since logs have temporal properties, we considered their time spans, frequencies (per hour, daily, and weekly), and start/end times as well as their impact factors that reflect the density of given events. For example, the impact factor of time t , IP_t can be calculated as Eq. (1), where x represents the event and $\alpha(x)$ and $\beta(x)$ represent the increment/decrement constants and monitoring time-span for event x . We set the value $\alpha(x)$ as 1 for each event x , and the values for $\beta(x)$ are set manually as follows; $\beta(\text{GPS}) = 1 \text{ h}$, $\beta(\text{Call}) = 1 \text{ h}$, $\beta(\text{SMS}) = 20 \text{ min}$, $\beta(\text{Pic. View}) = 5 \text{ min}$, $\beta(\text{Photographing}) = 30 \text{ min}$, and $\beta(\text{Mp3 Playing}) = 30 \text{ min}$,

$$IP_t(x) = \begin{cases} IP_{t-1}(x) + \alpha(x), & \text{if event } x \text{ is occurred} \\ IP_{t-1}(x) - \alpha(x), & \text{if event } x \text{ is not occurred in } \beta(x) \\ & \text{after prior event } x \end{cases} \quad (1)$$

The coordinates from the GPS log are used to get place names. In this paper, we divided the domain area into a lattice and then labeled each region. The user profiles and PIMS (personal information management system) datasets were used to find the user's social position (student, worker), gender, the position of their home, and the names and phone numbers of their friends and relatives.

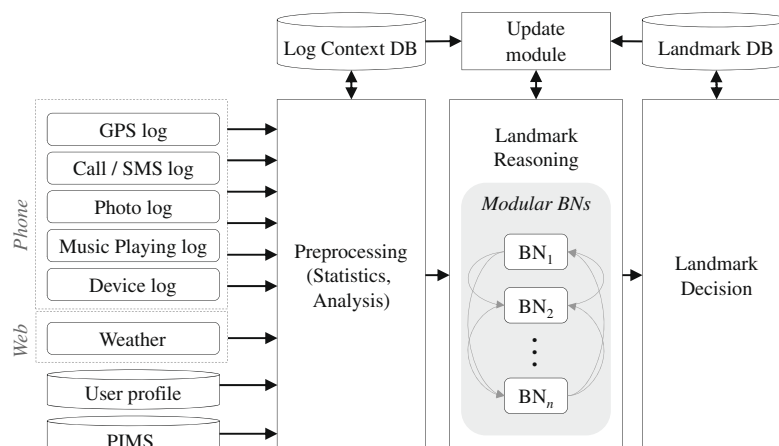


Fig. 1. The process of the landmark extraction from mobile log data.

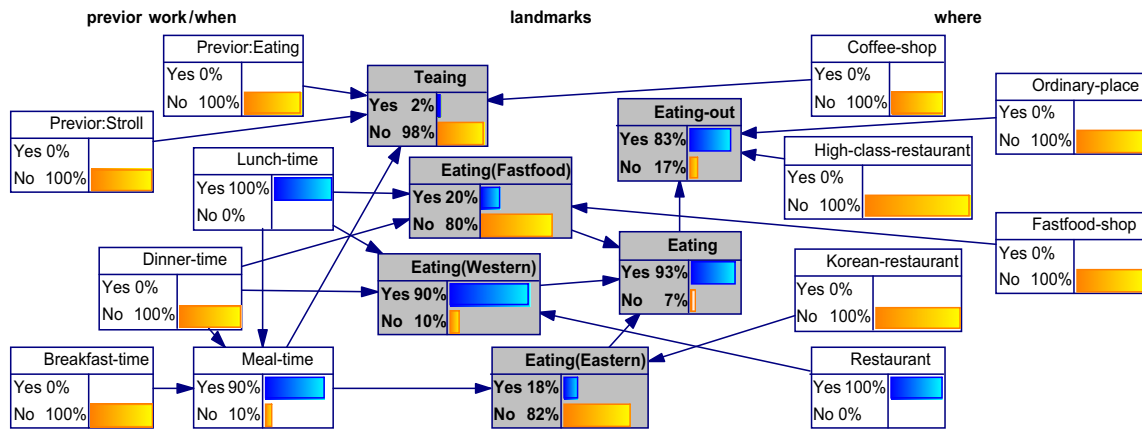


Fig. 2. The landmark inference BN 'Activity in restaurant' designed and used in this paper. This is 1 of 39 BNs designed. The node of gray box is landmark node. This shows the case that the 'Lunch-time' and 'Restaurant' evidence are set.

Table 1
The log information that was collected on a mobile device.

Log	Information
GPS	Latitude, longitude, velocity, direction, date, time
Call	Caller's phone number, call/receive/absence log, time span, start/end time
SMS	Sender's phone number, call/receive/absence log, time span, start/end time
Photographing	Photo file name, taking time
Weather	Weather, visibility range (km), cloud degree (%), temperature (°C), discomfort index, effective temperature (°C), rainfall (mm), snowfall (cm), humidity (%), wind direction, wind velocity (m/s), barometer (hPa)
MP3 Player	Title, time span, start/end time
Charging	Charging status, time span, start/end time

2.2. Cooperative reasoning of modular Bayesian networks

There are two general differences between the proposed BNs and conventional BNs. Firstly, we modularized the Bayesian inference models according to their separated sub-domains (Fig. 3). The required computing power for dealing with BN models is proportional to the number of nodes and arcs. Especially, since the computational complexity of Bayesian inference is approximately proportional to $O(\gamma^N)$, where γ is the number of states and N is the number of causal (parent) nodes, the modularized BN is more efficient.

Secondly, to consider the co-causality of the modularized BN, the proposed method shows 2-pass inference stages (Fig. 3). A vir-

tual linking technique is utilized to reflect the co-causal evidence more correctly. The technique is performed to add the virtual nodes and regulate their conditional probability values (CPVs) to apply the probability of the evidence. The process of the virtual linking technique is shown in Figs. 4 and 5. In Fig. 5, the virtual linking technique replaces given probabilistic evidence to the prior probability or conditional probability of virtual node value to keep the connectivity of BNs.

For example, when the structure of BN_1 is $\{A \rightarrow B \rightarrow C\}$, BN_2 is its modularized version $\{A \rightarrow B, B \rightarrow C\}$, and A and C are evidence and result nodes, we can calculate the belief value of node C using the virtual link assumption and chain rule such as Eqs. (2)–(5)

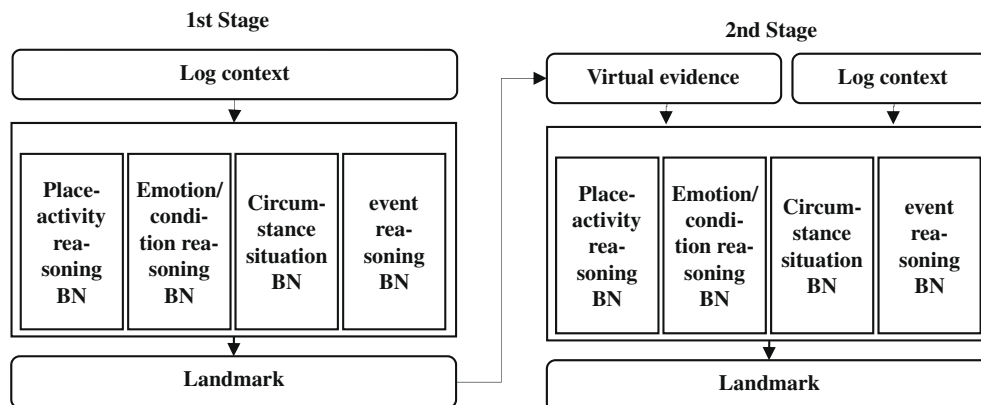


Fig. 3. The 2-pass inference process for the cooperation of modular Bayesian networks. It uses the result from inference of Bayesian network as the evidences at the 2nd inference stage. The parenthesis indicates the number of BNs contained. There are four kinds of BNs, and totally 39 BNs are used. The BNs are as follows: 19 place-activity BNs = {houses, religion, shopping, photographs, hospitals, nature, meetings, workplaces, sports, movements, food, calls, music, schools, traffic, amusement, busy, watching, resting}, 10 emotion/condition BNs = {joy, hunger, hot temperatures, cold temperatures, nonsense, surprise, tiredness, drunk, anger, worry, gloom, sickness, boredom}, 5 circumstance situation BNs = {space, climate, time, device, group}, 2 event BNs = {anniversaries, other events}.

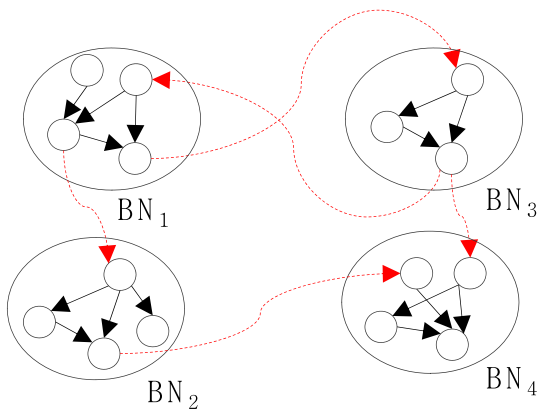


Fig. 4. Virtually linked Bayesian networks. Dotted line denotes some nodes linked by virtual linking technique.

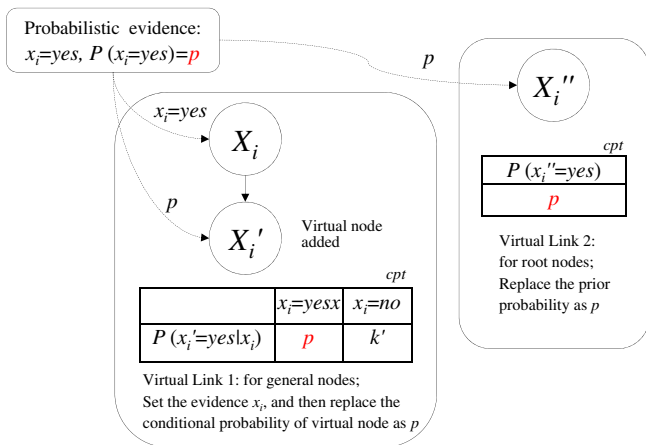


Fig. 5. The virtual linking method using virtual evidence enabling to use probabilistic evidences.

where the proposed virtual link assumption is $P(B) = Bel(B)$, and $Bel(B)$ means belief value of the node B (Korb & Nicholson, 2003),

$$BN_1's Bel(C) = P(A, B, C) = P(C | B)P(B | A)P(A) \quad (2)$$

$$BN_2's Bel(B) = P(A, B) = P(B | A)P(A) \quad (3)$$

$$BN_2's Bel(C) = P(B, C) = P(C | B)P(B) \quad (4)$$

$$\begin{aligned} &= P(C | B)Bel(B) \text{ since } P(B) \\ &= Bel(B) \text{ by virtual link assumption} \\ &= P(C | B)P(B | A)P(A) \end{aligned}$$

$$\therefore BN_2's Bel(C) = BN_1's Bel(C) \quad (5)$$

From the cooperative reasoning process we can get the landmarks and their probability distribution. Fig. 6 shows a pseudo code for the landmark extraction process.

2.3. Landmark decision

The final landmark decision is obtained by the belief probability, thresholds and weights of each landmark node after reasoning the modular BNs. The threshold is used to tune the landmark extraction model. Generally, the training data is not uniform enough to obtain an almost-perfect model, so we need to tune the model. Eq. (6) shows the computations to obtain the selected landmark set LM_s , where b_i represents the belief probability of node X_i , w_i is the weight of node X_i , and LM is the set of landmark nodes. The weights are used to apply the preferences or life-patterns of the given user, so if the landmark is a preferred by an user, the

- Preprocess the life log data L , and Generate Contexts
- For all BNs – 1st step of inference
 1. Initialize BN_m
 2. Set all evidence contexts for BN_m
 3. Update Belief of BN by running the inference algorithm
 4. Keep the landmarks and their probability values for virtual evidence
- For all BNs – 2nd step of inference
 1. Set all virtual evidences for BN_m
 2. Update Belief of BN by running the inference algorithm
 3. Save the landmarks and their probability values
- Select and decide extracted landmark set with threshold and weight

Fig. 6. The pseudo codes for landmark extraction processes. L is log data set.

weight has higher value than others and the landmark can be selected more easily,

$$LM_s = \{X_i | b_i \cdot w_i > th_i \wedge \forall X_i \in LM \wedge 0 \leq w_i \leq 1\} \quad (6)$$

2.4. Complexity analysis

To compare the complexities of the proposed modular BNs and the ordinary BN, we combined the designed BNs into a BN as shown in Fig. 7. A total of 39 BNs were designed with 638 nodes, 623 arcs and 4205 CPVs, as summarized in Table 2. On average, 16.6 nodes and 107.8 CPVs are used at the same time for the BN reasoning process since they are modularized and the computations are distributed. However, even though the combined BN for them has 462 nodes, which is decreased from 638 after removing the duplicated nodes on the combining process, the number of parents and CPVs are increased. This means that it has much larger complexity.

Eq. (7) shows the time complexity of the exact inference of the BN using the Lauritzen Spiegelhalter (LS) algorithm (Lauritzen & Spiegelhalter, 1988), which is the most popular exact inference algorithm and a junction tree-based algorithm, where 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 clique that each parameter used in the LS algorithm (Namasivayam & Prasanna, 2006),

$$cmpx_{inf} = O(k^3 n^k + wn^2 + (wr^w + r^w)n) \quad (7)$$

We have replaced the maximum clique size w with k , since the clique size is proportional to the parents' size, and the number of values r is about 2 in this paper, so Eq. (7) was reduced as shown by Eq. (8).

$$cmpx_{inf} \cong O(k^3 n^k + kn^2 + 2^k(k+1)n) \quad (8)$$

The ideal modularization technique roughly divides the number of nodes by the number of modules d but it has to compute d times for d modules, so the exact inference complexity of the modular BNs is as shown in Eq. (9) where the new number of nodes is (n/d) . As shown in Table 2, the proposed modularization technique roughly divides the number of nodes n by 27.8 and the number of parents by 1.3, so the exact inference complexity is decreased significantly,

$$cmpx'_{inf} \cong O\left(\frac{k^3}{d^{k-1}} n^k + \frac{k}{d} n^2 + 2^k(k+1)n\right) \quad (9)$$

3. Learning modular Bayesian networks from data

In this section we introduce a method to learn the proposed modular BNs automatically from the training data set.

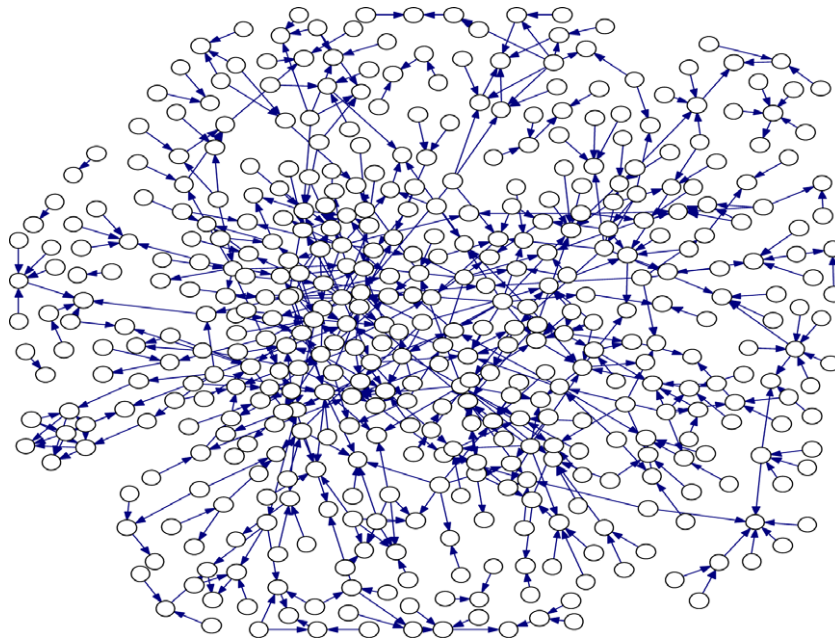


Fig. 7. The complex BN structure combined from the 39 designed BNs. This has 588 arcs and 462 nodes. The graph is drawn by GeNIe 2.0 (Graphical BN tool).

Table 2

The 39 modular BNs and the monolithic BN of them. MonoBN – the monolithic BN corresponding to the modular BNs, NN – no. of nodes, NNR – no. of root nodes, NNI – no. of intermediate nodes, NNL – no. of leaf nodes, NL – no. of arcs, NP_{avg} – avg. no. of parents, NS – no. of states, NS_{avg} – avg. no. of states, NCPV – no. of CPVs, CPT_{max} – maximum size of CPT.

	NN	NNR	NNI	NNL	NL	NP _{avg}	NS	NS _{avg}	NCPV	CPT _{max}
39 BNs	638	375	135	128	623	0.98	1279	2.00	4205	64
MonoBN	462	235	111	116	588	1.27	927	2.01	4.869	512

3.1. Learning Bayesian network from data

The BN model G can be defined as (B_s, Θ) , which means a network structure B_s and a probability parameter set Θ . $\Theta = \{B_\phi, B_p\}$ is composed of the conditional probability table B_ϕ and the prior probability distribution B_p . In general, learning Bayesian networks from data consists of two parts. The first is to learn the structure B_s and the other part is to learn the parameter set. Learning the structure can be solved by searching for the network structure which best matches the given data. The fitness can be measured by some scoring metrics. Two popular metrics are the minimum description length (MDL) score (Lam, 1998) and the Bayesian Dirichlet (BD) score (Heckermann, Geiger, & Chickering, 1995). The BD score for the structure B_s given the training data set D is defined as Eq. (10). To find the best structure with score metric, greedy search algorithms can be used. Fig. 8 shows a general process for the score-metric based structure learning,

$$BD(B_s; D) = P(B_s, D) = P(B_s)P(D|B_s) = P(B_s) \int P(D|\theta, B_s)P(\theta|B_s) d\theta \quad (10)$$

Parameter learning can be cast as either estimating the most likely parameter values (maximum likelihood learning) or updating the posterior probability distributions of the parameters given data (Bayesian learning). The parameter θ is calculated from the training data set D by using Eq. (11).

$$\theta^* = \arg \max_{\theta} P(D|\theta)P(\theta) \quad (11)$$

$P(\theta)$ means prior probability. The general discovery process is shown by Eq. (12) where $Z_T = \{z_1, z_2, \dots, z_T\}$ represents a set of T status variables, and Y_T is a set of T observation result variables,

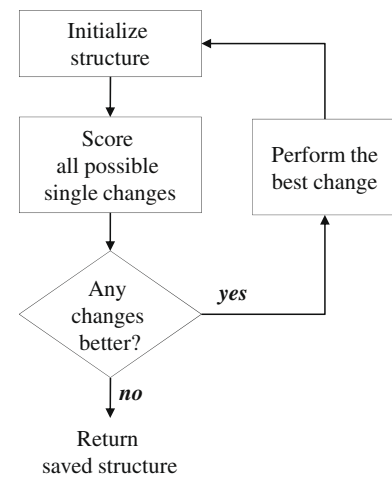


Fig. 8. A general process for score metric based structure learning.

$$P(Z_T, Y_T, \theta) = P(Y_T|Z_T, B_\phi)P(Z_T|B_p) \quad (12)$$

The conditional probability table B_ϕ is calculated from the relational frequency between the observation values and the status variables, meaning the probabilistic histogram of the training data.

3.2. Discovering the modular Bayesian networks

The proposed learning process of the modular BNs is shown in Fig. 9. This method includes a parameter setting process (for node

set X , the topological order set of nodes O , a level set of nodes V and the maximum size of parents p that each node had), a structure learning (for B_c) and a parameter learning (for θ). The node set of BN (X) is composed of the union of the collected log context set (L) and the target landmark set (LM). The values of the landmark set are defined by an expert user. The topological order set of nodes (O), the level set of nodes (V), and the maximum size of the parents (p) have been prepared for structure learning (the proposed method; domain-constrained K2 learning). To determine the topological order (O), we ranks each variable based on its total influence on the other variables with mutual information. The domain-constrained K2 learning method is proposed to constrain the parent domain and the hierarchical levels of the nodes. The learning method and the mutual information computation are presented in the next section.

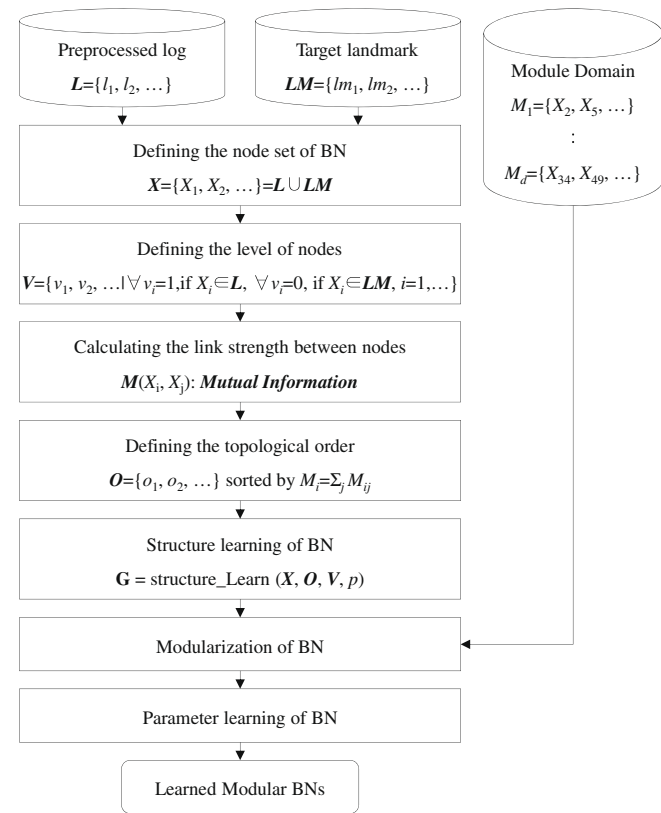


Fig. 9. The learning procedure of modular Bayesian network.

1. initialize $\{G_1, G_2, \dots, G_d\}$ based on $\{M_1, M_2, \dots, M_d\}$ and L
 $G_k = \{X_i | X_i \in G \wedge X_i \in M_k\}, k=1, 2, \dots, d$
2. make the arcs in $\{G_1, G_2, \dots, G_d\}$ based on G
 $G_k = G_k \cup \{arc_{ij} | \forall X_i, \forall X_j, \{X_i, X_j\} \subset G \wedge \{X_i, X_j\} \subset G \wedge arc_{ij} \in G \wedge i \neq j\}, k=1, 2, \dots, d$
3. add the virtual nodes in $\{G_1, G_2, \dots, G_d\}$ based on G
 $G_k = G_k \cup \{arc_{i'j}, X_{i'} = \text{virtual node of } X_i, \forall X_i, \{X_i, X_j\} \subset G \wedge X_i \in G_m \wedge X_j \in G_k \wedge X_i \notin G_k \wedge X_j \notin G_m \wedge arc_{ij} \in G \wedge i \neq j \wedge m \neq k\}, k=1, 2, \dots, d$

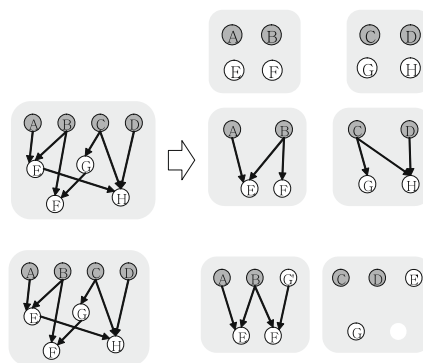


Fig. 10. The modularization process of BN. The right side shows an example, in which the gray nodes denote evidence nodes and the white nodes mean landmark nodes or their virtual nodes.

The network structures learned are divided into several modules by a modularization process as shown in Fig. 10, where G is the network structure and d denotes the number of modules. Process 1 defines the nodes of the module-BNs based on the module domain and the log context set. Process 2 defines the arcs of the module BNs based on the network structure (G). Process 3 defines the virtual nodes based on the network structure (G). The virtual node usage used for virtual linking between the BNs is described in Section 2.2.

3.3. Domain-constrained K2 algorithm

The domain-constrained K2 method proposed in this paper is based on our prior work (Hwang & Cho, 2004) and aims to learn the structure of the BN hierarchically with a constrained parent domain of nodes. The algorithm limits the hierarchical level at which a node can be positioned and reduces the search domain for the parent of the nodes. It can control the general direction of causality and decrease the searching complexity. In this paper, we use the algorithm to exclude the arcs between the evidence nodes (but we permits arcs between the landmarks) and maintain the direction of the arcs in order to use the virtual link technique. Table 3 shows the level and domain settings used in this paper.

Fig. 11 shows the proposed domain-constrained K2 algorithm, based on the K2 algorithm as proposed by Cooper and Herskovits (1992), which is the most popular BN algorithm and the basis of many advanced learning algorithms. This algorithm adopts a score metric known as the K2 metric, which calculates scores based on the difference between the BN graph and the training data distribution. The search process of the K2 algorithm is greedy and heuristic, and the K2 metric is calculated using Eq. (13),

$$P(G, D) = P(G) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (13)$$

where r_i represents the number of status variables, x_i , q_i means the number of combinations of variables included in π_i , N_{ijk} denotes the count in the dataset D , π_i is the j th combination, and x_i is the k th

Table 3
The level and domain definition.

Level	Node domain	Parent domain
0	$LM = \{lm_1, lm_2, \dots\}$	$D_0 = \Phi$
1	$L = \{l_1, l_2, \dots\}$	$D_1 = L \cup LM$

LM is landmark set, L is log context set.

```

Sort_nodes_by_topological_order(X, O)
for i=1 to n do:
     $\pi_i = \phi$  // initialize the parent set
    Score[i]=g(xi,  $\pi_i$ ) // k2 metric score of node xi
    Continue=true
    repeat
        Z= arg maxj g(xi,  $\pi_i \cup \{x_j\}$ ) and j<i and xj  $\notin$   $\pi_i$  and xj  $\in$  Dlevel(i)
        Score'[i]= g(xi,  $\pi_i \cup \{x_z\}$ )
        if Score'[i]>Score[i] then
            Score[i]= Score'[i],  $\pi_i = \pi_i \cup \{x_z\}$ 
        else Continue=false
    until | $\pi_i$ |<MaxParentNum and Continue=true
    
```

Fig. 11. The domain-constrained BN structure learning algorithm. D_k denotes the parent domain of kth level, and level(i) means level of x_i and MaxParentNum is a limitation of the number of parents.

value. The N_{ij} value is computed by Eq. (14) and the function $g()$ is calculated by Eq. (15),

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk!} \tag{14}$$

$$g(x_i, Pa(x_i)) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk!} \tag{15}$$

The K2 algorithm uses a topological order to maintain the graph as the DAG by maintaining that the prior node cannot be the child of the posterior node without any other DAG checking rules. However, we have to optimize the topological order since a different topological order will lead to a different BN structure. In this paper, we compute the influence score of all the nodes by using mutual information (Su & Zhang, 2006), and sort out the topological order with the score. Eqs. (16) and (17) show the influence score and the mutual information calculation,

$$M_i = \sum_j^{M(X_i, X_j)} \tag{16}$$

$$M(X; Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \tag{17}$$

3.4. Threshold and weight

The weight value presents the importance of landmarks and can be set by humans in terms of their preferences. In this paper, we set all the weight values as 1.0 to observe the general performance of landmark extraction.

The performance of a classifier depends on its threshold since it determines the result status of the node. Especially in BN classifiers, it is changeable because the belief probability distribution is based on the number and distribution of training data values, so we have defined the threshold as the value that shows the best hit rate for the training data. Fig. 12 shows an example of the classification results of the node 'moving.' In the results, the hit rate

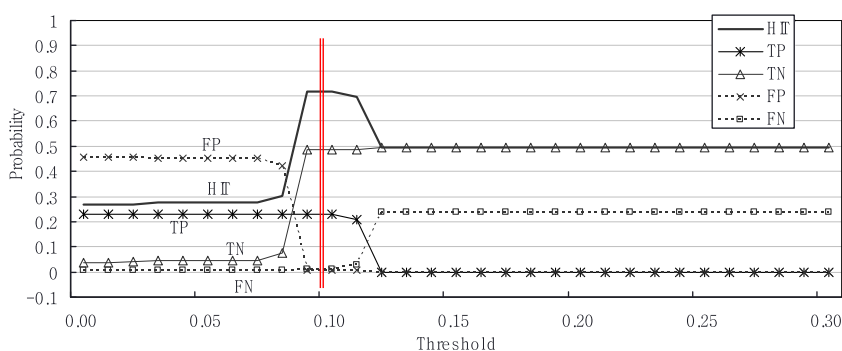


Fig. 12. The classification performance of the node 'moving' with threshold transitions (from a modular BN trained with the parameter $p = 4$). The rest threshold (more than 0.3) is omitted since there is no change. The value 0.1 shows the best performance.

Table 4

The category, id and names of nodes.

Category	IDs	Node names
Landmark nodes (48)	N001, N002, ..., N048	Moving, using vehicle, school-club activity, eat out, lecture, sleeping, surprising, fret, joy, throb, tired, meet friend, cold, meal (Korean), meet family, busy, hungry, drinking (alcohol), take a walk, troublesome, overflowing joy, bored, ride a vehicle, weight-training, home activity, with leisure, hair-cut, study, yearning, school activity, traffic jam, run, walk, go to school, meal (western), eat (tee), ready to go out, date with my date, sad, test in school, sleepy, meet kin, employment counsel, extracurricular lecture, late for school, concentration, eat (Chinese), supper
Log context nodes (110)	N049, N050, N051, ..., N156, N157, N158	SMS sender = unknown, take a picture, listening music, calling with kin or friend, receive a call, listening music for a long time, charging, good scenery place, GPS is operating, average velocity is less than 20 km/h, speed is from 5 km/h to 20 km/h, amusement grounds, velocity is from 20 km/h to 150 km/h, shopping place, lecture building, velocity is less than 5 km/h, town, front of school, department store, rest place, bus station, soccer play ground, park, traffic place, a post office, street, public institution, home, 0 o'clock, 1 o'clock, 2 o'clock, ..., 23 o'clock, night, dawn, morning, AM, PM, evening, daytime, sunshine-time, week, calling, ignore of call, velocity is faster than 150 km, out of town, merrymaking place, subway station, tourist resort, terminal of exp. bus, express bus, student building, place for meal, church, place for special meal, school, coffee shop, out of town, front of YS Univ., main load of YS Univ., front of lecture building, main entrance of school, wedding place, museum, street basket-ball court, coffee shop, Japanese restaurant, playground, library, lecture building, theater, middle school, weather = fine, Thursday, spring, frequent call, many call, electronic and electric products center, commercial center, subway station, Restaurant, many photo, high-school, Dong-dae-moon place, weather = fine, Hospital, dining room, Korean restaurant, house of kin, house of friend, Chinese restaurant

The parenthesis denotes the number of nodes.

Table 5
Module domain definition of landmark nodes.

Domain name	Number	Landmarks
Emotion and status	17	Bored, busy, cold, concentration, fret, hungry, joy, overflowing joy, sad, sleepy, surprising, throb, tired, troublesome, with leisure, yearning
Everyday life	14	Eat (Chinese), eat (western), eat (Korean), home activity, meet family, moving, ready to go out, ride a vehicle, run, sleeping, supper, using vehicle, walk
Event	9	Date with my date, drinking(alcohol), eat (tee), eat out, hair-cut, meet friend, meet kin, take a walk, traffic jam, weight-training
School life	9	Employment counsel, extracurricular lecture, go to school, late for school, lecture, school activity, school-club activity, study, test in school
Total	48	

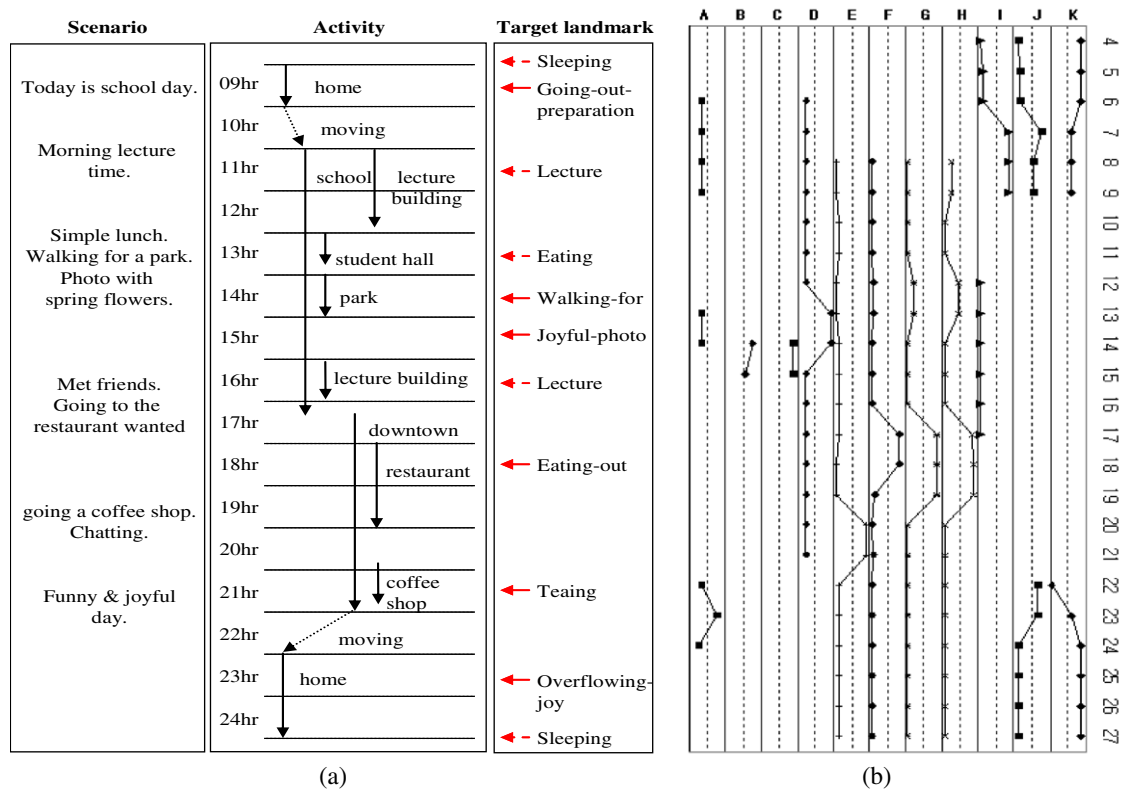


Fig. 13. (a) A scenario of an everyday life with mobile device of an undergraduate student for experiments. (b) The observation of the probability values of 11 target landmarks. The denoted time is from 4 o'clock to 27 o'clock (equal to 3 o'clock of the next day). Abbreviations used for landmarks: A – overflowing-joy, B – photo (scenery), C – joyful-photo, D – walking-for, E – tea, F – eating-out, G – eating (western style), H – eating, I – going-out-preparation, J – shower, K – sleeping.

was the best when the threshold was 0.1, and we adopted the value for the threshold of the nodes. This procedure was conducted automatically during BN training stage.

4. Experiments

In this section, we analyze the proposed method with experiments with generated and collected data. We evaluate the reasoning performance of the landmark extraction model designed and learned. Then, we observe the experimental results. Finally, we compare the auto-trained modular BNs and their monolithic versions.

4.1. Experimental data

The log data used in this paper include GPS log, call log, SMS log, picture log, music-playing log, device charging log, and weather log obtained from a website. The total number of designed BNs is 39 (19 activity reasoning BNs for 19 kinds of places, 13 emotion/con-

dition reasoning BNs for the users, 5 circumstances or situation BNs, and 2 event reasoning BNs).

We collected data from three college students (women) with smartphones for a month. These users performed subtasks (such as writing activity diaries, shopping, walking and calling) to make the data more useful.

The experimental data was segmented into units of ten minutes, so there were 2304 entries. Redundant data were excluded resulting in 779 examples remained. We defined 48 landmarks and used 110 life log contexts (as evidence) as shown in Table 4. To learn the modular BNs, we divided the landmarks into four modules based on four categories (Emotion and status, Everyday life, Events, and School life) as shown in Table 5.

4.2. A case study

We tested the proposed landmark-reasoning model with the following scenario in order to confirm performance. The left side of Fig. 13 shows the scenario used. The BN set strongly related to

Table 6

The real life log data collected in smartphone. We have excluded the data of unstable GPS log. N_{Con} – the number of input contexts (evidences), N_{LM} – the number of extracted landmark candidates, N'_{LM} – the number of selected landmarks, N'_{HIT} – the number of hitting landmarks, N_{LM} – the number of partial hitting landmarks, N_{ERR} – the number of landmarks, R_{HIT} – the rate of N_{HIT} , R'_{HIT} – the rate of N'_{HIT} , R_{ERR} – the error rate.

Date	N_{Con}	N_{LM}	N'_{LM}	N_{HIT}	N'_{HIT}	N_{ERR}	R_{HIT}	R'_{HIT}	R_{ERR}
02-24	116	72	13	3	10	0	23.1	100.0	0.0
02-27	167	49	15	4	11	0	26.7	100.0	0.0
02-28	64	50	8	3	4	1	37.5	87.5	12.5
<u>03-02</u>	202	128	18	8	10	0	44.4	100.0	0.0
<u>03-04</u>	102	53	7	1	5	1	14.3	85.7	14.3
<u>03-06</u>	86	56	12	5	3	4	41.7	66.7	33.3
<u>03-08</u>	114	92	12	3	7	2	25.0	83.3	16.7
<u>03-09</u>	103	45	7	4	2	1	57.1	85.7	14.3
03-15	128	76	13	4	9	0	30.8	100.0	0.0
03-17	46	45	8	3	3	2	37.5	75.0	25.0
03-21	67	40	10	4	4	2	40.0	80.0	20.0
Total	1195	706	123	42	68	13	34.1	89.4	10.6

the scenario is {food, photo, movement, nature, joy, home}. The probabilities were calculated when the related evidence was given. The log context data used as evidence includes 26 artificial log contexts: going out (in 2 h), lecture-building, daytime, restaurant, photography, eating, entertainment, before breakfast, condition, morning, going-out (before), moving, daylight-hours, sleeping (in

2 h), nature, device-in-use, dinner(before), good-weather, joyful day, home, charging, coffee-shop, ordinary-place, on campus, student-hall, GPS-running.

After generating the log contexts during the day, we tested them. The right side of Fig. 13 shows the inference results. We can see the high probability values of the related landmarks at the corresponding times. For example, there are 'going-out-preparation' and 'shower' landmarks at 7–9 o'clock, 'eating' at 12–13 and 17–19 o'clock, 'walking' at 13–14 and 20–21 o'clock, 'photography' at 14–15 o'clock, and 'eating' landmarks at 17–19 o'clock.

4.3. Performance evaluation of the designed BNs

To evaluate performance of the designed BNs, we operated our landmark extraction model. Table 6 shows the results of landmark extraction using data from eleven days collected from human1. The data of the underlined date was also used in the learning experiment. In the experiment, we set all landmark thresholds as 66% to avoid worthless landmarks, since a human designed the BN models by focusing on true positive relations naturally and without regarding negative relations between the logs and the landmarks.

Since the log data were uncertain and the activity diary did not cover all possible life contexts, we evaluated the results (RHIT') using a wide viewpoint (permitting partial hits), which means that the landmarks could be reasonably detected. We decided the

Table 7

The matching results of extracted landmarks. They are evaluated based on the activity diaries and GPS trajectories of users.

Date	Hit landmark	Partial hit landmark	Missing (omission)
02-24	Meal (Korean), meeting, shopping	Studying, lecture, lecture time, busy, unpleasant SMS, viewing, eat out, traffic jam, singing, dancing at club	–
02-27	Study, joyful photograph, meeting, shopping	Lecture time, busy, unpleasant SMS, photograph (food), photograph (product), photograph (scene), viewing, meal (Korean), eat out, singing, dancing at club	–
02-28	Busy, joyful call, meeting	Lecture time, unpleasant SMS, viewing, studying	Shopping
03-02	Lecture time, busy, joyful call, meal (Korean), eat out, meeting, take a walk	Unpleasant SMS, photograph (food), photograph (product), photograph (scene), viewing, disappointment, shopping, singing, dancing at club, studying, studying till late	–
03-04	Viewing	Computer working, washing face, busy, unpleasant SMS, traffic jam	Lecture time
03-06	Meeting, unpleasant SMS, lecture time, busy, meal (Korean)	Take a walk, viewing, traffic jam	Eat out, shopping, singing, dancing at club
03-08	Lecture time, meal (Korean), eat out	Unpleasant SMS, busy, traffic jam, meeting, shopping, singing, dancing at club	Viewing, take tea
03-09	Study, meeting, lecture time, viewing, meeting	Busy, traffic jam	Shopping
03-15	Lecture time, meal (Korean), eat out, shopping	Cleaning, cooking, dishwashing, busy, viewing, meeting, singing, dancing at club, take a walk	–
03-17	Studying, take a walk, lecture time	Busy, disappointment, meeting	Shopping, viewing
03-21	Study till late, lecture time, meal (Korean), shopping	Busy, meeting, singing, dancing at club	Viewing, eat out

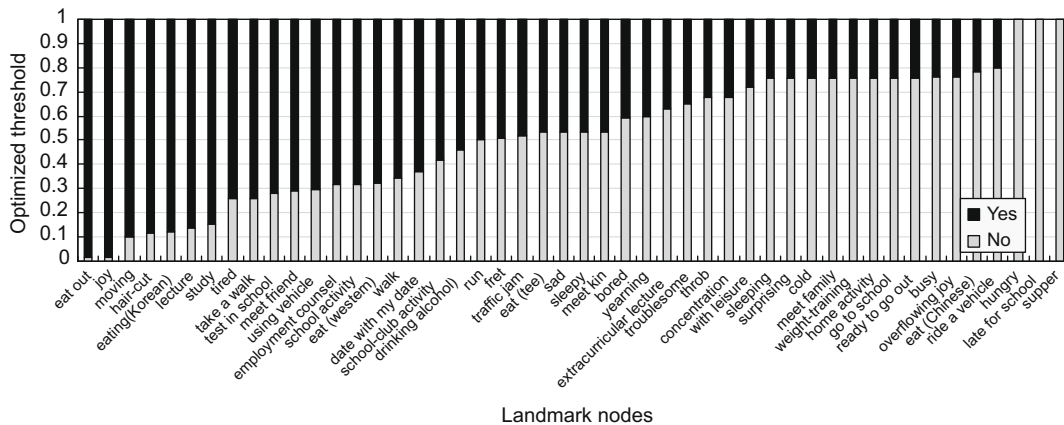


Fig. 14. The optimized threshold set for node. The thresholds are sorted by the value.

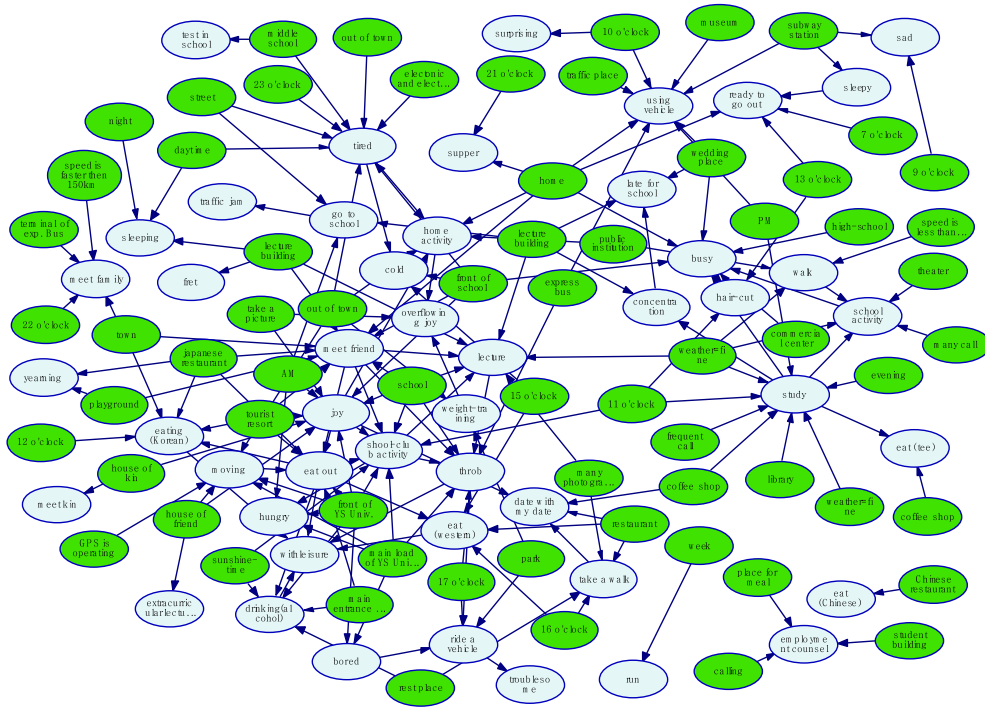


Fig. 15. The monolithic BN trained with parameter $p = 8$. The brighter nodes are landmark nodes and the others are evidence nodes.

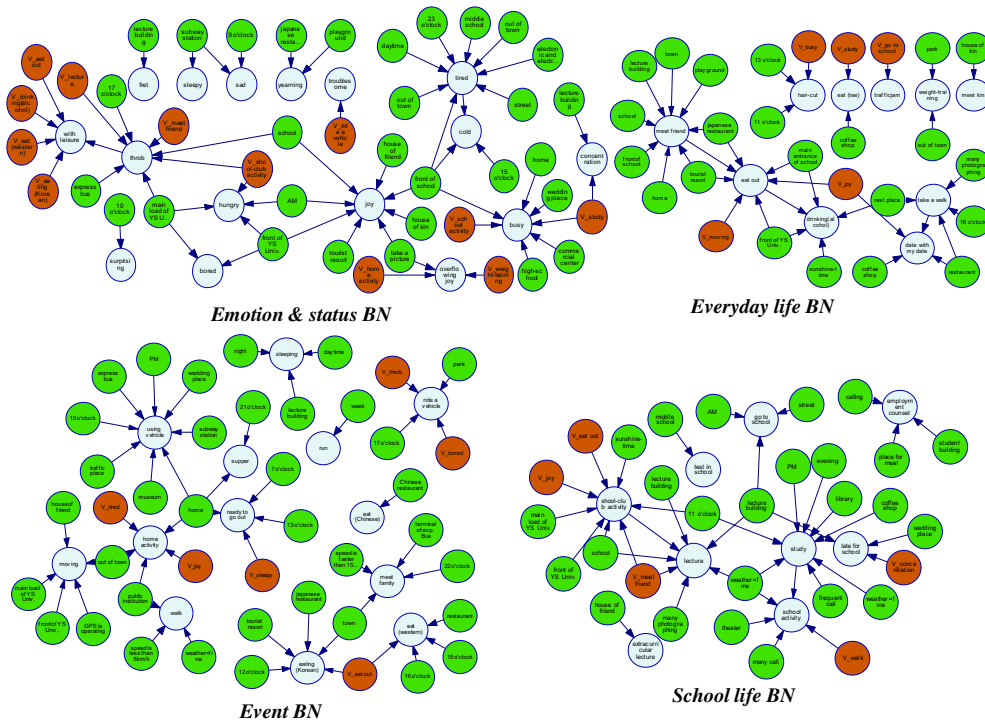


Fig. 16. The BN modules trained using the proposed learning method with parameter $p = 8$. The dark nodes: virtual nodes, the lighter nodes are landmark nodes, and the others (medium brightness) are evidence nodes.

partial hits by majority of 5 persons. As a result, the rate of the complete hit landmarks (RHIT) was mostly low (34.1% in total) but the rate of the partial hit landmarks (RHIT') was as high as 89.4%. Especially, when the activities were plenty, the results were better since the contexts were sufficient. Table 7 shows the specific results of landmark reasoning. For example, the user's real activity on 03-02 was practicing music until late at night, but the land-

mark 'studying till late' was detected. We classified landmarks like this as partial hit landmarks.

4.4. Performance evaluation of the trained BNs

In this section, we describe the test result of the landmark extraction model using 16 days of mobile life log data obtained

by the proposed modular BN learning method. We set the MaxParentNum parameters (p) as 4 and 8 in the experiment. The automatically-discovered threshold values for the landmarks of the modular BNs are shown in Fig. 14, where the landmarks 'hungry', 'late for school', and 'supper' cannot be detected when the threshold is 1. This means the training data do not contain the related evidences. The other landmarks show various threshold values.

Figs. 15 and 16 show the monolithic BN and modular BNs learned with (parameter $p = 8$). The average number of nodes, parents, and conditional probability values and the level of complexity are shown in Table 8. The complexities are calculated by Eq. (8) and we can observe the decrement of the complexity of the modular BNs.

Table 9 shows the results of the landmark reasoning evaluation. Because the number of training data was small, we used the leave-one-out validation method. We compared the monolithic BN and modular BNs with the parameters $p = 4$ and $p = 8$. The computation of the precision rate is $(TP/(TP + FP))$, and the hit rate is $((TP + TN)/(TP + TN + FP + FN))$.

As shown by the results, the performance of the modular BNs is similar to that of the monolithic BN. The measured values are same when the parent size parameter p is 4. This means the proposed method is valuable since it is used to reduce the BN model and increase efficiency.

The results with $p = 4$ is better than with $p = 8$. We think that the bigger p increase complexity by allowing more parents and cause lack of training. How many parameters of BN are not trained

Table 8

The comparison of the number of nodes, parents, and conditional probability values and the complexity of learned BNs. The complexities are calculated based on Eq. (8).

BN	Nodes #	Nodes # _{avg}	Parents #	Parents # _{avg}	Complexity
Mono BN	115	115	298	2.59	$O(2.6 \times 10^6)$
Modular BNs	160	40	182	1.14	$O(0.388 \times 10^6)$

– the number of, #_{avg} – the average number of.

Table 9

The correctness comparison of extracted landmarks. Same values are underlined.

BN	P	TP	TN	FP	FN	Precision	Hit rate
Mono BN	8	135	35,845	64	1,348	0.678	<u>0.962</u>
Modular BNs	8	133	35,853	56	1,350	0.704	<u>0.962</u>
Mono BN	4	<u>420</u>	<u>35,807</u>	<u>102</u>	<u>1,063</u>	<u>0.805</u>	<u>0.969</u>
Modular BNs	4	<u>420</u>	<u>35,807</u>	<u>102</u>	<u>1,063</u>	<u>0.805</u>	<u>0.969</u>

TP – true positive, TN – true negative, FP – false positive, FN – false negative.

Table 10

The comparison of the execution time on Pocket PC emulator.

BN	Environment	Number of BN	Execution time (clk)
Modular BNs 1	Pocket PC	39	100
combined BN	Pocket PC	1	180
Modular BNs 2	Pocket PC	4	50
Monolithic BN	Pocket PC	1	N/A – out of memory error

from data can be measured by the number of CPVs that is 0.5 since an initial untrained CPV is 0.5. The number of the trained mono BN with $p = 4$ is 582 and with $p = 8$ is 4396, so it is reasonable that the BN with $p = 4$ is better than with $p = 8$.

4.5. Comparison of the execution speed

In order to compare the computation complexities between the ordinary monolithic BN model and the proposed model, we have tested their execution time on mobile device simulation. The subjects are designed 39 BNs, their monolithic BN, trained BN, and their monolithic BN. In this experiment, since we used 2-step inference method for modular BN model, the reasoning process is conducted twice for each BN. We conducted 10 times of inference for each BN model. The test environment for mobile device is as follows:

- Simulation Tool: Microsoft Pocket PC 2003 SDK.
- Operating System: Pocket PC 2003.
- Memory: 44 MB.

Table 10 shows the experimental results. The execution time denotes CPU clock time-span during 10 reasoning processes. The monolithic and combined BN took more time than modular BNs on Pocket PC environments, but some experiments with monolithic BN failed to run because of out-of-memory error. This means that the ordinary BN cannot handle the landmark detection task on the constrained environment like pocket PC where the domain is too large while the proposed method can manage the task well.

To find the executable size of BN for monolithic BN 2 (for the learned), we tested the BN with progressive deletion of nodes. Fig. 17 shows the experimental result. The learned monolithic BN could operate after deleting the 3rd node. It means that the 3rd node requires much computation complexity. Actually, the 3rd node had 11 parents and has 4096 conditional probability parameters. It can be understood that some complex nodes cause much computation and make troubles.

In the experiments, we can observe that the proposed model reduced the computation complexity of BN and enabled to handle more complex probabilistic model on the constrained environment even when the original BN model cannot work.

5. Concluding remarks

In this paper, we proposed a modular landmark inference model, which was more efficient and suitable to mobile environments. We introduced the modularized BN model for efficient operations in mobile environments, and proposed the 2-step inference method by applying the virtual node concept, and then learned the modular BNs automatically from the given training data. In experimental results with artificial and real mobile life log data, the intended landmarks were well-extracted and the proposed method was able to reduce the level of complexity. We also discussed how to define the learning parameters and thresholds

However, in this paper, we did not sufficiently cover the temporal properties of human landmarks. In the future, we need to continue research using a dynamic BN model that manages temporal features

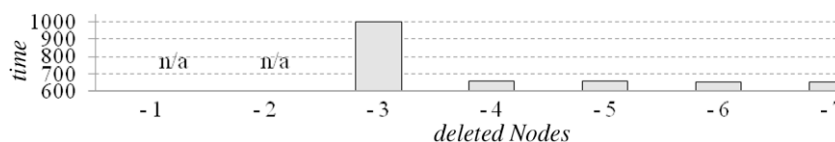


Fig. 17. The execution time of the learned monolithic BNs 2 with node deletion on Pocket PC emulation.

well. Also, experiments with sufficient real world data should be conducted for a longer period of time.

Acknowledgement

This work was supported by MKE, Korea under ITRC IITA-2009-(c1090-0902-0046) and UCN 09C1-T3-11T.

References

- Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.
- DeVaul, R., Sung, M., Gips, J., & Pentland, A. (2003). MITHril 2003: Applications and architecture. In *Proceedings of the 7th IEEE international symposium on wearable computers* (pp. 4–11).
- Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1), 19–30.
- Gemmell, J., Williams, L., Wood, K., Lueder, R., & Bell, G. (2004). Pervasive capture and ensuing issues for a personal lifetime store. In *Proceedings of the 1st ACM workshop on continuous archival and retrieval of personal experiences, October* (pp. 48–55).
- GeNle 2.0. Decision Systems Laboratory, University of Pittsburgh. <<http://genie.sis.pitt.edu>>.
- Heckermann, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3), 197–243.
- Horvitz, E., Dumais, S., & Koch, P. (2004). Learning predictive models of memory landmarks. In *CogSci 2004: 26th annual meeting of the cognitive science society* (pp. 1–6).
- Hwang, K.-S., & Cho, S.-B. (2004). Constrained learning method of Bayesian network structure for efficient context classification. *Proceedings of Korea Information Science Society*, 31(2), 112–114 (in Korean).
- Hwang, K.-S., & Cho, S.-B. (2006). Modular Bayesian networks for inferring landmarks on mobile daily life. In *The 19th Australian joint conference on artificial intelligence* (pp. 929–933).
- Hwang, K.-B., Kim, B.-H., & Zhang, B.-T. (2006). Learning hierarchical Bayesian networks for large-scale data analysis. *International Conference on Neural Information Processing*, 1, 670–679.
- Ji, Q., Lan, P., & Looney, C. (2006). A probabilistic framework for modeling and real-time monitoring human fatigue. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 36(5), 862–875.
- Korb, K. B., & Nicholson, A. E. (2003). *Bayesian artificial intelligence*. Chapman & Hall/CRC.
- Korpipaa, P., Mantjarvi, J., Kela, J., Keranen, H., & Malm, E.-J. (2003). Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3), 42–51.
- Krause, A., Smailagic, A., & Siewiorek, D. P. (2006). Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array. *IEEE Transactions on Mobile Computing*, 5(2), 113–127.
- Lam, W. (1998). Bayesian network refinement via machine learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3), 240–251.
- Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 157–224.
- Li, X., & Ji, Q. (2005). Active affective state detection and user assistance with dynamic Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(6), 93–105.
- Lo, B. P. L., Thiemjarus, S., & Yang, G.-Z. (2003). Adaptive Bayesian networks for video processing. *International Conference on Image Processing*, 1(1), 889–892.
- Marengoni, M., Hanson, A., Zilberstein, S., & Riseman, E. (2003). Decision making and uncertainty management in a 3D reconstruction system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), 852–858.
- Namasivayam, V. K., & Prasanna, V. K. (2006). Scalable parallel implementation of exact inference in Bayesian networks. *International Conference on Parallel and Distributed Systems*.
- Nokia LifeBlog. <<http://www.nokia.com/lifeblog>>.
- Raento, M., Oulasvirta, A., Petit, R., & Toivonen, H. (2005). ContextPhone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2), 51–59.
- Schmidt, A., Takaluoma, A., & Mntyjrv, J. (2000). Context-aware telephony over WAP. *Personal Technologies*, 4(4), 225–229.
- Siewiorek, D. P., Smailagic, A., Furakawa, J., Krause, A., Moraveji, N., Reiger, K., et al. (2003). SenSay: A context-aware mobile phone. In *Proceedings of the 7th international symposium of wearable computers, October* (pp. 248–249).
- Su, J., & Zhang, H. (2006). Full Bayesian network classifiers. In *Proceedings of the international conference on machine learning* (pp. 897–904).
- Tu, H., Allanach, J., Singh, S., Pattipati, K. R., & Willett, P. (2006). Information integration via hierarchical and hybrid Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(1), 19–33.
- Zhang, Y., & Ji, Q. (2006). Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(2), 467–472.
- Zheng, P., & Ni, L. M. (2006). The rise of the smartphone. *IEEE Distributed Systems Online*, 7(3).