PAPER

User-Adapted Recommendation of Content on Mobile Devices Using Bayesian Networks

Hirotoshi IWASAKI[†a)], ***Member***, **Nobuhiro MIZUNO**[†], **Kousuke HARA**[†], ***Nonmembers***, ***and*** **Yoichi MOTOMURA**[††], ***Member***

SUMMARY Mobile devices, such as cellular phones and car navigation systems, are essential to daily life. People acquire necessary information and preferred content over communication networks anywhere, anytime. However, usability issues arise from the simplicity of user interfaces themselves. Thus, a recommendation of content that is adapted to a user's preference and situation will help the user select content. In this paper, we describe a method to realize such a system using Bayesian networks. This user-adapted mobile system is based on a user model that provides recommendation of content (i.e., restaurants, shops, and music that are suitable to the user and situation) and that learns incrementally based on accumulated usage history data. However, sufficient samples are not always guaranteed, since a user model would require combined dependency among users, situations, and contents. Therefore, we propose the LK method for modeling, which complements incomplete and insufficient samples using knowledge data, and CPT incremental learning for adaptation based on a small number of samples. In order to evaluate the methods proposed, we applied them to restaurant recommendations made on car navigation systems. The evaluation results confirmed that our model based on the LK method can be expected to provide better generalization performance than that of the conventional method. Furthermore, our system would require much less operation than current car navigation systems from the beginning of use. Our evaluation results also indicate that learning a user's individual preference through CPT incremental learning would be beneficial to many users, even with only a few samples. As a result, we have developed the technology of a system that becomes more adapted to a user the more it is used.
key words: *adaptive interface, Bayesian network, situation awareness, mobile device, recommender system, user model*

1. Introduction

Mobile devices, such as cellular phones and car navigation systems, are essential to daily life. People acquire necessary information and preferred content over communication networks anywhere, anytime. While cellular phones naturally have access to providers, car navigation systems are also able to access telematics providers [1], [2] that provide the latest information on traffic, news, points of interest (POIs), and other practical data.

However, usability issues arise from the simplicity of user interfaces themselves. In addition to the small size of the devices, their input devices (e.g., touch screens) are simple, unlike devices that allow easy input for users (e.g., keyboards). During content retrieval, hierarchical searches are deep, making it difficult for users to reach their preferred content in a short time. For example, to search for a restaurant on a car navigation system, a driver often uses the category hierarchical search. The driver first chooses a category, such as "restaurant." Next, the driver chooses a cuisine, such as "Japanese" or "French." The system then lists the names of 200 nearby restaurants. The driver finds a preferable restaurant by scrolling the list or refining the search based on various attributes. If no preferable restaurants appear, the driver changes the cuisine choice and starts again.

For easier selection from much information, the simplest method would be to have the user preliminarily set up favorite content for every situation. However, many users would not complete the set-up, since it would be too troublesome. In contrast, if a system could automatically recommend content that is adapted to a user's preference and situation, the user would be able to acquire content easily. This user-adapted mobile system must satisfy three requirements. First, it should not require extra operation for adaptation, since content selection is a secondary task for mobile users. Practically, a user selects content as a usual operation, with no extra procedure (e.g., many settings and evaluation of the content). Second, the recommendation should be situation-aware, considering such criteria as when, where, and with whom. For example, preference for a restaurant might depend on the time of day. With respect to the situation, a car navigation system knows many kinds of user situations, such as calendar information, current position, and passenger. A cellular phone with a Global Positioning System (GPS) can detect the same information. Lastly, the system must adapt to each individual user, since content preference differs for each individual user and may change over time. Moreover, it is necessary for the system to consider the privacy of the individual user. In addition, it must adapt under the limitation of the number of training samples from the individual user and the limitation of computational resources of mobile devices.

In this paper, we describe a method to realize a user-adapted mobile system using Bayesian networks (Fig. 1). This system recommends content such as restaurants, shops, and music suitable to the user and the situation. It learns the model incrementally based on accumulated usage history data, with no extra operation. The more it is used, the more it becomes adapted to the user.

Manuscript received June 10, 2009.
Manuscript revised November 22, 2009.
[†]The authors are with DENSO IT LABORATORY, INC., Tokyo, 150–0002 Japan.
[††]The author is with Center for Service Research, The National Institute of Advanced Industrial Science and Technology, Tokyo, 135–0064 Japan.
a) E-mail: hiwasaki@d-itlab.co.jp
DOI: 10.1587/transinf.E93.D.1186

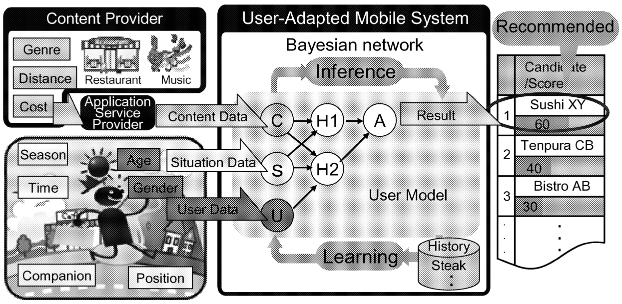

Fig. 1 User-adapted mobile system.

This paper is organized as follows. Section 2 briefly reviews some related work on methods of recommendation. Section 3 explains the proposed methods for modeling a user model and adapting the model, using Bayesian networks. In Sect. 4, we implement the recommendation of restaurants in order to evaluate our methods. Section 5 discusses the evaluation and its results. Section 6 summarizes our work.

2. Related Work

Recommendation methods can be classified into two families. The first is collaborative filtering [3], [4], in which a system makes predictions about suitable items, based on feedback from many different users. GroupLens [3] is an early example of this family. The system filters Usenet news using the nearest neighbor method. News content is filtered based on evaluation results of other users who have performed the same evaluation as the user. This technology has been widely adopted by many web sites such as Amazon.com, which has a centralized recommender system of books and CDs for sales promotion. However, privacy issues arise when a provider collects users' activity histories. Many users probably would not want their activity history (e.g., time, place, and companion) to be known by others.

The other family is content-based filtering, in which a system describes content based on the features of the content as well as models user evaluations of each feature of the content, and recommends content by comparing the two. Syskill & Webert [5] is one of representative examples of this family. The system recommends a web page in which a user indicates interest in a certain topic. Based on the evaluated degree of satisfaction with the browsed page, the system estimates the degree of preference for the page in the search results. An evaluation of six recommendation techniques revealed that the Bayesian Classifier is the most efficient.

In order to take a situation into account, several approaches are considered in this family. The approaches are divided into three categories. The first category is a rule-based approach [6]. A rule is described in detail for every situation, as in "content A is recommended in situation X." The cost of rule establishment and correction is high since the number of combinations of contents and situations considered is enormous. Moreover, it is impossible to apply content in a situation that has no rule established.

The second category is a memory-based approach. The nearest neighbor algorithm is most typical [7], [8]. All training samples that include user evaluations are simply stored in memory. In order to classify a new content, the algorithm compares it with all stored samples, using a similarity function, and determines the nearest neighbor or the k nearest neighbors. The system recommends content based on the evaluations of the neighbors. A situation can be introduced into a similarity function. However, the computational cost is high, due to the large number of samples and high dimension of the training samples including the situation. The system would take too much time to make a recommendation.

The last category is a model-based approach. One of the most typical approaches uses a neural network [9]. A neural network models a nonlinear phenomenon that involves learning a multilayer perceptron through back propagation. A user's preference can be modeled by mapping content suitable for a preference to a situation. However, this approach is not desirable for adaptation due to the stability-plasticity dilemma in which incremental data may cause a network to forget completely all previous training samples that would not be stored in a mobile device.

Recently, a Bayesian network [10] has drawn researchers' attention as a technique for user modeling [11]–[13]. A Bayesian network models uncertain phenomena, such as preferences and situations. Some of related works are satisfying the first two requirements. For example, the Bayesphone [11] is a portable device that handles a call to a user, considering the situation by learning a user model from samples provided by the user in advance. Ono et al. [13] proposed recommendations of movies that consider the user's situation (e.g., companion), based on a Bayesian network model constructed using a large-scale web-based questionnaire and introductory texts on movies. These systems are based on large samples. However, in the user-adapted mobile system, sufficient samples are not always guaranteed since the user model requires combined dependency among users, situations, and contents. Moreover, these methods do not tackle adaptation in accordance with individual users.

3. Modeling Method

In this section, we first review the general modeling method of the Bayesian network, and then present our proposed methods.

3.1 Bayesian Network

A Bayesian network is a graphical knowledge model expressing probabilistic knowledge in a directed non-cyclic graph. It consists of a graph structure that qualitatively indicates the dependency between variables and the conditional probability that quantitatively indicates it. A graph is comprised of nodes and links. A node that sends a link to another node is called a parent node, and the other is called a child node. The user model has many discrete values. When a variable is a discrete value, conditional probability is expressed by a Conditional Probability Table (CPT). In order to construct a Bayesian network model, it is necessary to determine CPTs and a graph structure. When training samples are complete, the CPT can be learned from samples; particularly with a large amount of samples, the CPT can be estimated by maximum likelihood estimation. However, the samples from users and situations expressing the situation tend to be incomplete and insufficient. In contrast, the determination of the graph structure from samples is NP-complete [14]; various local search algorithms, such as the K2 algorithm [15], have been proposed. The K2 algorithm reduces computational effort by a greedy search as follows.

1. A total order relation along the parent-child relation of all nodes is defined for reduction of search space.
2. A child node is selected.
3. A parent node set is assigned to the child node according to the total order relation.
4. The CPT of the child node with one parent node from the set is calculated in order for every parent node in the set.
5. Best combination is selected according to the information criterion, which is a measure of the goodness of an estimated model.
6. The number of parent nodes is increased and continued.
7. When the best model is found, the following child node is selected, and this algorithm is repeated.

This algorithm is effective when sufficient samples are acquired. However, it is not easy to acquire enough samples on the user-adapted mobile system. In the mobile environment, various kinds of situations (e.g., when, where, and with whom) should be considered for recommendation. Therefore, it is necessary to acquire a huge number of samples from which a combination of attributes of users (e.g., age and occupation) and situations (e.g., time, kind of next schedule, and passenger's pattern) differs. To overcome this issue, the Bayesian network has the advantage of being able to assign a meaning to each node and being readable by designers. Thus, the domain knowledge of a domain expert can be put into a graph structure as partial order relations and into CPTs as prior probabilities. With the use of such knowledge, the generalization performance of the system is expected to be high.

We propose a method that merges samples and domain knowledge to complement incomplete and insufficient samples. For example, if we know dependency between age and cuisine, we can connect their nodes by altering bounds of states of their CPT. We call this method the "learning a model using domain knowledge" (LK) method (Fig. 2).

3.2 LK Method

1) Requirement definition: The object to be modeled is clarified by defining the user's requirement for a system as a "Use Case." The Use Case is a notation for defining system behavior in the Unified Modeling Language (UML) [16], which is widely adopted in software engineering.

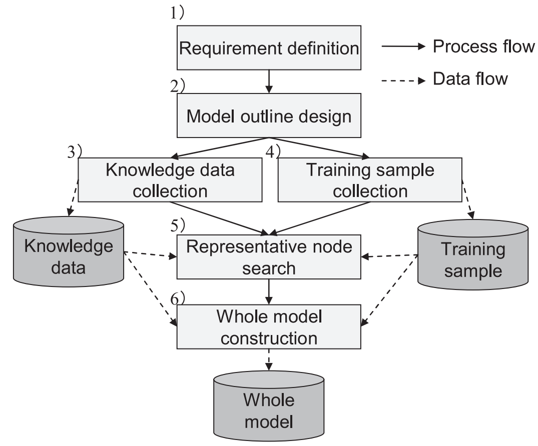

Fig. 2 LK method.

2) Model outline design: An outline of the model structure is designed by analyzing the Use Case. This process determines criterion variables and explanatory variables. Variables are grouped according to semantic dependency, and dependency between groups is defined.
3) Knowledge data collection: Knowledge data are collected from the domain knowledge of experts, users, and designers. Data are stored in the form of partial order relations and prior probabilities, which represent dependency of variables.
4) Training sample collection: Training samples are collected from users by such means as questionnaires and records of content selection. When the collection is complete, record recount, data cleansing, and data complement are important for high prediction accuracy.
5) Representative node search: Based on the model outline design, a partial model is constructed in each group unit, and representative nodes are searched. A partial model is learned from training samples and constructed out of the variables in each group. A representative node is an independent node representing the information about a group. It consists of the top parent node of each partial model, and the independent node. Using these nodes, statistical independence can be improved. Knowledge data are used for altering the model in this process.
6) Whole model construction: The whole model is constructed by combining partial models focusing on representative nodes. Knowledge data are also used in this process.

The following subsections present details of the representative node search and the whole model construction.

3.2.1 Representative Node Search

The representative node search consists of two parts (Fig. 3). The first half is a process for narrowing the parent node candidates for each child node in the variables V_n in a group. The second half is a process for searching for parent node candidates for each child node and constructing partial mod-

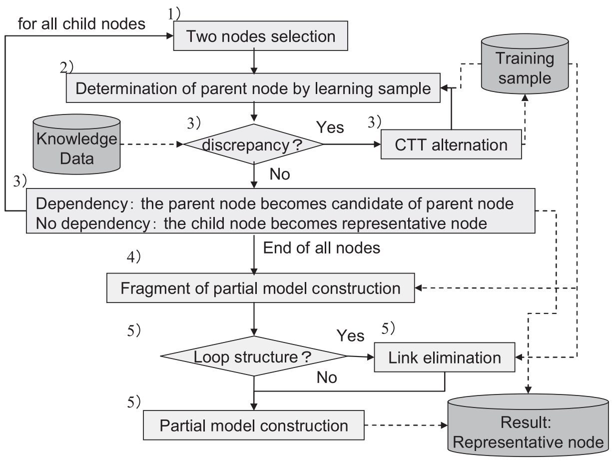

Fig. 3 Representative node search.

els.
1) Two-node selection: For a child node $X_i \in V_n$, a parent node X_k is chosen from a parent node set $pa(X_i) = \{X_k | V_n \backslash X_i\}$, and a one-to-one model $B(X_k \rightarrow X_i)$ is constructed, where the arrow means direction of a link.
2) Determination of parent node by training samples: Using training samples and normalizing the cross-tabulation table (CTT), conditional probabilities of a CPT are learned, and an information criterion IC is evaluated. If the following inequality is true, the parent node X_k is a parent node candidate.

$$IC(B(X_k \rightarrow X_i), S) < IC(B(X_i), S), \tag{1}$$

where S represents samples, the left-hand side expresses the information criterion of the child node, and a lower value of information criterion represents the better model.

In addition, prior probabilities collected as knowledge data are set in the CPT of the child node X_i.
3) Knowledge data for determining node dependency: The dependency between the two nodes is compared with the knowledge data that specifies a partial order relation. If the existence of the dependency has a conflict, the CTT of the child node is altered, and dependency is evaluated again. The alteration involves aggregation with other nodes, division of a node, and division and integration of the component of a state in a CTT. The altered CPT is checked against an information criterion, and judgment is made whether or not to adopt the alteration. If the alteration is not effective, dependency follows the dependency acquired from the training samples, noting that the knowledge data contains vague information. The parent node that has dependency becomes a member of a new parent node set $pa_1(X_i)$. If the child node has no parent node candidate, the node becomes a representative node.
4) Fragment of partial model construction: To a child node, the parent node set $pa_2(X_i)$ that becomes best on an information criterion is searched in a set $pa_1(X_i)$ by starting from

$$X_1^* = \arg\min_k IC(B(X_k \rightarrow X_i), S). \tag{2}$$

This algorithm is like the K2 algorithm. This process is repeated for each child node that did not become a representative node. Finally, the fragment of the partial model of two levels is made for the number of the child nodes.
5) Partial model construction: These fragments are composed simply, and some partial models are constructed. With the availability of a structure through which a link circulates, links that have less influence of evaluation on an information criterion are deleted [17].

3.2.2 Whole Model Construction

The whole model construction follows almost the same procedure as the representative node search. The only difference is that the combination of nodes to search in the first half is the combination of nodes between groups with dependency designed by the model outline.
1) Dependency search between two groups: The one-to-one dependency between representative nodes of each of the two groups is searched. After resolving any conflict with knowledge data, a parent node with dependency becomes a parent node candidate. If no parent node with dependency exists, the search range for a parent node is extended to the child node of the representative parent node, and the search begins again. This process lasts for all child nodes of a partial model.

Furthermore, if no parent node with dependency exists, the range the child node searches is extended to the child node of the representative parent node, and the search is repeated. This process continues for all child nodes of a partial model.
2) Dependency search for all groups: This search is carried out among all the groups with dependence, and the first half is ended.

3.3 Adaptation Method

When an individual user begins to use the user-adapted mobile system with the initial model that is constructed by the LK method, the system incrementally and automatically learns the model from each individual user's history of content selection and activities (Fig. 4). Adaptation is effective for the user who does not have a high-precision recommendation at the beginning of use, as well as for the user whose preference changes over time.

The adaptation method is a learning method basically, which is the same as the modeling method. It consists of structural learning and CPT learning. This naïve approach stores all previously seen data, and repeatedly invokes a batch learning procedure after each new sample is recorded. Though this approach can learn an optimal model, it requires a huge amount of computational power and memory. To solve this problem, several approaches have been proposed [18]. These approaches avoid storing all previously seen data, and partially search a semi-optimal structure. However, it is difficult for the embedded system, which is used in the mobile environment, to provide enough com-

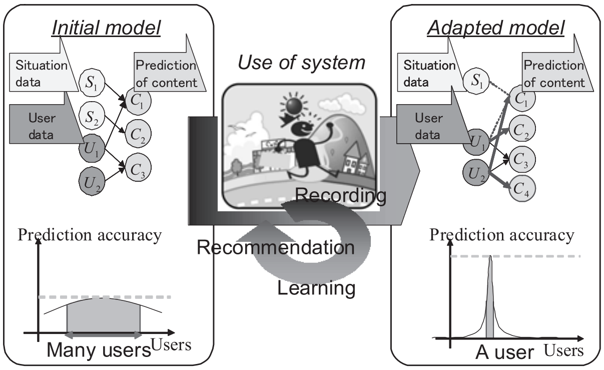

Fig. 4 User adaptation.

putational resources for structural learning. It is also difficult to collect enough samples from one user to perform structural learning, because the recommendations are not frequent. Therefore, incremental learning is accomplished by applying Bayesian learning to the CPT.

This approach regards the initial model as knowledge data. Since the initial model has a structure built in consideration of the compatibility of knowledge and samples, this approach is based on the idea that it probably has better structure for many users than the model constructed by a few samples at the time of user adaptation. In addition, the model based on this approach is expected to have high prediction accuracy even with a small number of samples since it makes use of the CPT of the initial model as prior probabilities. Moreover, this approach is applicable to a mobile system since it needs little computational power.

In this approach, fractional updating that simply performs Bayesian learning to the CPT using incremental data is proposed [19]. However, this method is not effective when incremental data is much less than data that are used for the initial model. Furthermore, incremental data can be regarded as representing the user's preference more precisely than the others' data that are used for the initial model.

We propose a method that presents a parameter, the scale-factor coefficient, indicating the importance of incremental data to the initial model. This parameter is multiplied by the incremental data. We call this method CPT incremental learning. The algorithm is as follows.

CPT Incremental Learning:

1. for all nodes
2. select a node
3. set up a CTT for the incremental data of the node
4. multiply the CTT of 3 by the scale-factor coefficient α, and add it to the CTT of the initial model
5. normalize the CTT as probabilities for the CPT.

The parameter depends on each problem domain and is determined by the number of incremental data and the variation of the individual user's preference from the general preference in the initial model.

4. User-Adapted Car Navigation System

To evaluate our proposed methods, we applied our methods to restaurant recommendations on a car navigation system. The user-adapted car navigation system has user models and recommends content such as restaurants suitable for a user preference and situation. It needs the set-up of user attributes (e.g., date of birth and disposal income) by a user only at the beginning of use. Also, it learns the models incrementally, based on accumulated usage history data, through normal operation (e.g., selecting a restaurant and going to a restaurant).

4.1 Outline of a User Model

To construct a high-precision model, we adopted insights from psychology regarding preference. One model that describes customer preference is the multi-attribute attitude model [20]. According to this model, the decision to purchase articles and services is made by evaluating many attributes. This model is expressed as

$$A_i = \sum_{j=1}^{n} e_j b_{ij}, \tag{3}$$

where A_i is the attitude (i.e., degree of preference) toward the content i, n is the number of attributes rated, e_j is the weight of attribute j, and b_{ij} is the evaluative aspect toward attribute j. This model predicts that the content with the highest value of this attitude is the most preferred. However, the predicting capability of the multi-attribute attitude model is limited since a person is subjected to information overload when too many attributes exist. Therefore, various decision-making strategies that involve a series of mental operations for evaluation and decision-making on alternatives have been proposed. Two classifications of representative strategies exist. One involves compensation: the compensatory models have compensation between attributes, whereas the non-compensatory models do not. The other addresses the search process. The attribute-processing models represent the process of comparing the attributes of each alternative, while the brand-processing models focus on the process of comparing each alternative. In actual decision-making, individuals combine more than one model.

A user model consists of a multi-attribute attitude model and a Bayesian network model (Fig. 5). The Bayesian network model infers b_{ij}, and the multi-attribute attitude model calculates an attitude score. As a result, the following equation is derived:

$$A_i = \sum_{j=1}^{n} e_j \log p(C_j = c_{ij}), \tag{4}$$

where $\log p(C_j = c_{ij})$ is a logarithm likelihood of attribute j of content i inferred by the Bayesian network.

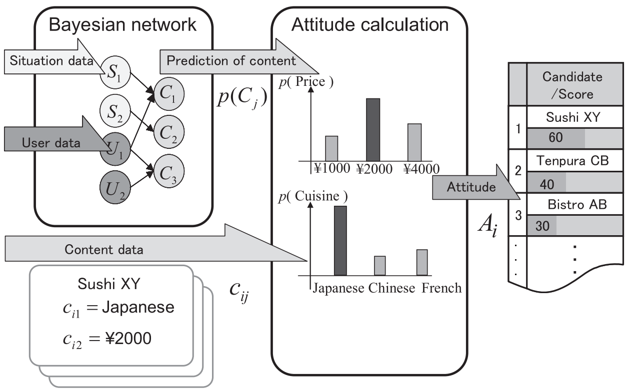

Fig. 5 Outline of a user model.

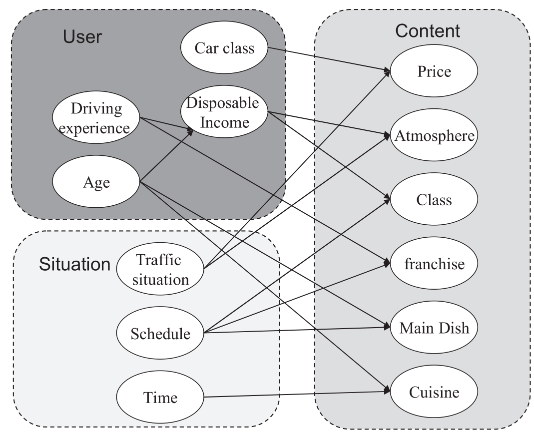

Fig. 6 Restaurant-preference model.

4.2 Constructed Bayesian Network Model

Figure 6 illustrates a restaurant-preference model, which is a Bayesian network in a user model built using the LK method with domain knowledge data and samples from a web questionnaire.

Knowledge data were collected from six general users. These users were selected so that their attributes might vary. They were men and women who drove frequently, and they were in their 20 s to 50 s. The group included office clerks, engineers, a manager, and a housewife. We asked dependency between the variables within user attributes, situation attributes, and content attributes, as well as dependency of the variables between user attributes and content attributes, and the variables between situation attributes and content attributes. The collected results were totaled, and when more than half of the users answered "related", the partial order relation was taken into account. In order to indicate the level of this test, let the null hypothesis be that there was no dependency, namely the probability of the population judging dependency "related" was less or equal 0.5, since this judgment contained only two possible results, "related" or "unrelated". The level of test, which was the probability of rejecting the null hypothesis, was 0.34. We made this level somewhat loose, since it was better to be judged "related" for knowledge data. The judgment would be revised using training samples; additionally, unnecessary dependency would wither through incremental learning by adjusting its CPT.

The questionnaire was submitted to 300 people who were presented with 6 out of 18 situations and asked to select a maximum of 3 restaurants from 182 candidates. In consideration of recommendation in a car, we divided the class of vehicle type into roughly the same number, and selected subjects so that the numbers of those in each class would be equal. The candidates were restaurants around Shinagawa Station in Tokyo. These restaurants were selected so that attributes might differ. We asked for 12 user attributes and presented situations with 12 attributes and restaurants with 17 attributes (Table 1). The user attributes were determined by adding attributes related to driving to demographic attributes. The other attributes were determined in interviews of eight men and women in their 20 s to 50 s; if the attribute was mentioned it became a criterion for restaurant selection. In the selection procedure in the questionnaire, the subject was first asked to indicate a preferred cuisine suitable for the situation from seven kinds, and was shown a list of the restaurants of the cuisine. If there were no preferable restaurant, the procedure asked for the cuisine again. This procedure is comparable to the category hierarchical search, which is the typical selection method of current car navigation systems. Accordingly, we acquired 3,778 samples.

Table 1 Attributes sought in the web questionnaire.

	Attributes
User	Age, Gender, Type of job, Driving experience, Annual income, Disposal income, Family structure, Car class, etc
Situation	Season, Day of the week, Time, Weather, Temperature, Area, Road category, Traffic situation, Schedule, etc
Content	Price, Atmosphere, Class, Franchise, Main dish, Cuisine, Smoking/non-smoking zone, Parking, Distance, etc

During model construction, we chose the information criterion based on similarity to knowledge data. Similarity was defined as the ratio of the number of incidences of joint selection of dependency as determined by knowledge data and information criterion to the total number of two-node sets. Specifically, using similarity, we chose the Akaike Information Criterion (AIC) [21] for the representative node search and Minimum Description Length (MDL) [22] for the whole model construction.

5. Evaluation

In this section, we describe the evaluation of the restaurant-preference model.

5.1 Evaluation of Model Construction

5.1.1 Evaluation Outline

To confirm the generalization performance of a model by the LK method (LK model), we compared it with a model by the K2 algorithm (K2 model). The generalization performance of the LK method has been improved using both domain knowledge and the training samples, even when the samples have bias due to insufficiency. Cross-validation that uses part of the samples as evaluation data is often applied for estimating the generalization performance by simulation. However, when the samples have bias, the evaluation data has the same feature. Since the prediction accuracy of the model that is constructed with only the samples (i.e., a K2 model) becomes higher due to the effect of bias, true generalization performance cannot be compared by cross-validation via simulation. Therefore, in order to compare the performance of two models, it is necessary to evaluate with different data from the samples. We constructed a K2 model, confirmed the hypothesis by simulation evaluation, and evaluated the generalization performance by user evaluation that is similar to a real user's operation.

In the K2 algorithm, a total order relation was determined from the lower stream (child node side) on the basis of each content node, situation node, and user node. A maximum of two parent nodes was searched for one child node, according to the LK model. In addition, the state of each node was made the same as that in the LK model.

In the simulation evaluation, we evaluated the prediction accuracy of the restaurant-preference model with a leave-one-out cross-validation, where "one" refers to a single subject's sample, using the records of the web questionnaire. Since crossing had little influence, we did not change the model structure, and we had the CPTs learn without the evaluation data of one subject. The evaluation criterion should be based on whether prediction works effectively, considering how a real user uses the system. A driver would have to select a restaurant after pulling over or while waiting for a traffic light to change, since operating a device while driving is prohibited for safety reasons. To evaluate the models, we defined prediction accuracy as the rate of acceptable recommendations. "Acceptable" meant that a restaurant could be selected within the waiting time for a traffic light to change (30 to 60 seconds). Rankings of 20th place and above were defined as "acceptable" ranking errors R, where ranking error was the ranking of a restaurant selected by a subject from the recommended list of restaurants.

$$PredictionAccuracy = |\{i\,|r_i \leq R\}|\,/\,N_{case}, \tag{5}$$

where r_i is the ranking error of case number i and N_{case} is the total number of cases.

In the user evaluation, we evaluated not only the prediction accuracy but also recall and "chosen method by subjects" as proof of the generalization performance. Recall is the rate of preferred restaurants within rankings of 20th place and above.

$$Recall_j = \left|\left\{k\,\middle|r_{jk} \leq R\right\}\right|\,/\,R, \tag{6}$$

where r_{jk} is the ranking error of restaurant k selected by subject j. Recall is averaged $Recall_j$ over subjects. "Chosen method by subjects" is the rate of subjects who chose a method as providing better recommendations.

$$Chosen\ method\ by\ subjects = n_{method}\,/\,N_{subject}, \tag{7}$$

where n_{method} is the number of subjects who choose the method and $N_{subject}$ is the total number of subjects. The evaluation was conducted with 15 subjects (11 men and 4 women) in their 20 s to 50 s, using a restaurant-recommendation application with a simple GUI. The application first asked for user attributes, then presented a situation and provided restaurants in recommended order from the same restaurants as were used in the questionnaire. The subject selected the best restaurant and some preferred restaurants. This evaluation was carried out with six different situations per subject. We asked subjects to select the best restaurant in 60 seconds, considering how a real user uses the system.

To investigate the differences between decision-making strategies, we evaluated the following two strategies via the simulation.

The additive rule: This compensatory brand model is the same as the multi-attribute attitude model.
The hybrid rule: This strategy uses the non-compensatory model first to filter important attributes, and then uses the compensatory model. This strategy is similar to the actual strategy.

The questionnaire asked subjects what restaurant attributes they considered necessary in selecting a restaurant. Using this information in the hybrid rule, we set the weight of each required attribute to 1; we set the others to 0. In contrast, in the additive rule, we set the weight of all attributes to 1.

We also evaluated the effect of our development by comparing it with current car navigation systems' category hierarchical searches. We used the result of the selection operation in the questionnaire, which was the same procedure as used in the category hierarchical searches. For comparison with current products, we considered the number of restaurants as ranking errors until a restaurant that a subject selected appeared since current products display restaurants as a list ordered by distance. We obtained this number from the result of the subject's selection in the same web questionnaire.

5.1.2 Evaluation Results

Figure 7 illustrates the result of the simulation evaluation, and Table 2 presents the result of the user evaluation.

Figure 7 depicts the frequency distribution of the ranking error acquired by 1800 individual evaluations (300 people × 6 situations) with respect to ranking error. Prediction accuracy is denoted by the leftmost rod, to all the frequencies. The evaluation result reveals the prediction accuracies of the additive-rule model (29 %), the hybrid-rule model (33 %), the current product (26 %), and the K2 model (46 %). The prediction accuracy of the K2 model was higher than that of the LK model, as expected (P-value = 0.000).

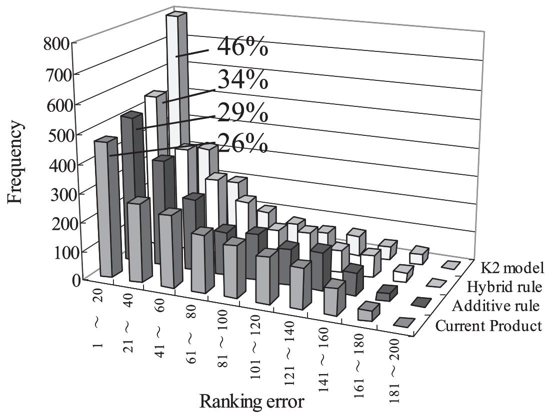

Fig. 7 Prediction accuracy of each model.

Table 2 User evaluation of LK method and K2 method.

	LK model	K2 model
Preciction accuracy	79%	67%
Recall	18%	14%
Chosen method by subjects	60%	33%

However, in Table 2, the user evaluation of the generalization performance indicates that the prediction accuracy of the LK model is higher (P-value = 0.045), in contrast to the simulation. Furthermore, all other criteria indicate that the LK model can produce better recommendations. As a result, we confirmed that the LK model has better generalization performance.

In addition, prediction accuracy of the user evaluation is high compared to the simulation evaluation. This is believed to result from increasing the preference of a restaurant due to the time limit of restaurant selection.

The poor evaluation of the K2 model in the user evaluation results from the extraordinary model (Fig. 8). It had no user node, and we could not explain the dependency. We presumed that the K2 model recommended more restaurants whose user dependencies were low, and whose chances levels were high in training samples; hence, its prediction accuracy was high. To confirm this assumption, we investigated the characteristics of the top ten ranked restaurant groups (ranked by the frequency that each restaurant was selected by subjects in the questionnaire) (Table 3). The number of selections in the groups accounted for 31 % of the total number of selections of each restaurant in the questionnaire (3778 records). The group included many chain restaurants that were comparatively easy for any subject to select. As expected, the group accounted for more than half (51 %) of the restaurants correctly recommended by the K2 model; the group accounted for 40 % of the restaurants correctly recommended by the LE model.

In addition, we compared the characteristics of recommendations by each model. Figure 9 indicates the rates of attributes of recommended restaurants 20th place and above. The recommendations were based on the same conditions as the questionnaire (1800 cases). These results indicate that the K2 model mainly recommended chain restaurants

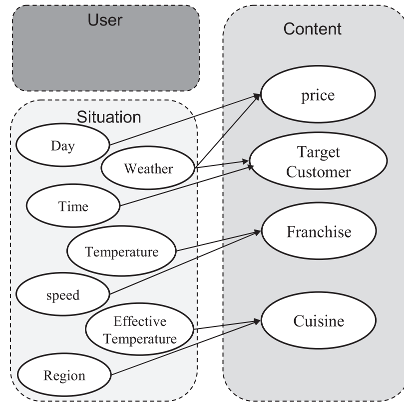

Fig. 8 K2 model.

Table 3 Feature of recommendation by LK method and K2 method.

#	Restaurant	Percentage of selections	Percentage of correct recommendation LK model	K2 model
1	Jitsuen teuchi udon Kineya	4%	17%	18%
2	STARBUCKS COFFEE	4%	0%	7%
3	Tenpura Tsunahachi SHINAGAWA INTERCITY-ten	4%	3%	1%
4	Ramen Kazuki Gotanda-ten	3%	0%	2%
5	Sinsyu sobadokoro Sojibo	3%	1%	5%
6	Spaghetti & Pizza24	3%	0%	1%
7	JONATHAN'S Higashi Gotanda-ten	3%	0%	4%
8	McDonald's	3%	1%	0%
9	Tonkatsu to Kaniwasyoku Inaba	3%	4%	4%
10	Tonkatsu Shichibee Shinagawa-ten	2%	14%	8%
	Total	31%	40%	51%

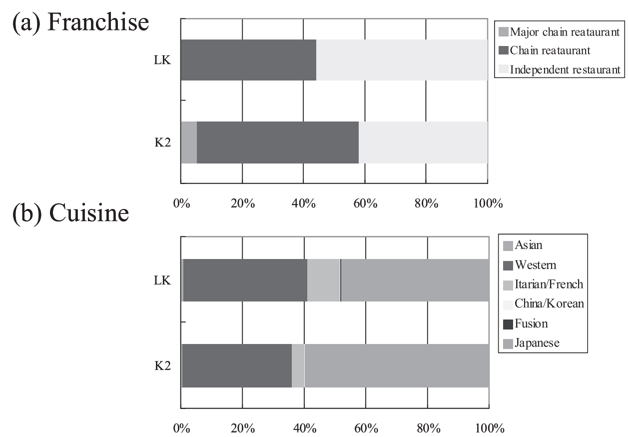

Fig. 9 Characteristics of recommendation by LK method and K2 method.

and Japanese restaurants. In contrast, the LK model recommended independent restaurants and various categories of restaurants. We presume that this difference is due to the existence of user nodes. In other words, there are not enough training samples in this evaluation to separate individual user preferences. Recommender system users would like to have a recommendation of a restaurant that suits their preference rather than a restaurant that is easy to find and exists anywhere. Therefore, the K2 model is not a truly precise model of generalization, even if it has high prediction accuracy in the simulation evaluation. Hence, knowledge data is useful when constructing a model from an insufficient number of samples.

The hybrid-rule model was more accurate than the additive-rule model. It also appeared to be closer to the actual decision-making strategies for restaurant selection. This result indicated the possibility of improving accuracy by predicting decision-making strategies, such as attributes of user requirements. However, the difference was not particularly large; thus, we will perform further evaluation, including other models of strategies.

The prediction accuracies of our systems are better than those of current car navigation systems. Furthermore, our system does not require selection of cuisine, so it requires less operation than current products. Even with first use, a driver using our system can select a suitable restaurant easier and faster as it considers many attributes.

5.2 Evaluation of User Adaptation

5.2.1 Evaluation Outline

To evaluate user adaptation, the results of the same web questionnaire were used for a leave-one-out cross-verification, where "one" refers to one sample of a subject. In the questionnaire, each subject selected at least one restaurant in each situation, so we obtained a minimum of 6 samples and a maximum of 18 samples from each subject. We used these samples to evaluate CPT incremental learning using the additive-rule model (corresponding to the second row of Fig. 7). For evaluation, we used each subject's prediction accuracy averaged over six recommendation results in six situations.

To confirm the effect of CPT incremental learning, we analyzed the prediction accuracy of the model with fractional updating.

We made the unit the ratio of the incremental data to all training samples (3,776 samples) to derive a scale-factor coefficient. We experimentally determined the coefficient by varying the ratio from 0.005 to 10.

We targeted user adaptation in three months (i.e., 24 usages, assuming two uses per week).

5.2.2 Evaluation Results

Figure 10 compares the prediction accuracy of CPT incremental learning to the ratio of the incremental data to all training samples. This prediction accuracy was averaged over all subjects (300 people) with five samples each. The ratio indicated high prediction accuracy (43 %) near 1. Based on this result, we set the ratio to 1 in subsequent evaluations.

Figure 11 plots the results of the average prediction accuracy of all subjects. Prediction accuracy improved from 29 % for initial use as the number of times used increased. As the number of uses increased, a monotonic rise of predictive accuracy was observed, and CPT incremental learning was effective even with a small number of samples. Furthermore, we extrapolated the results using second-order polynomial regression. After the subject used the system 24 times (assuming three months of use), prediction accuracy was 54 %. In contrast, prediction accuracy by fractional updating was 37 %. A monotonic but slow rise of predictive accuracy was observed. This result confirmed the effect of CPT incremental learning.

To evaluate the effect for each individual user, the prediction accuracy of the adapted model was compared to that of the initial model. Figure 12 presents the frequency distribution of subjects with respect to the prediction accuracy of each subject's adapted model. There are seven kinds of prediction accuracies, due to six results of recommendations. Each column in the stacked bar chart expresses the distribution of prediction accuracy of the initial model of each subject who had the same prediction accuracy of the adapted model. The adapted model was learned with five samples. The subject group whose results in the initial model were good (67 % or more) tended to become better after user adaptation (the two columns from the top in the 83 % and 100 % rods in comparison with the topmost columns in the 17 % and 33 % rods). These subjects were considered to have preferences near those of the initial model, which was

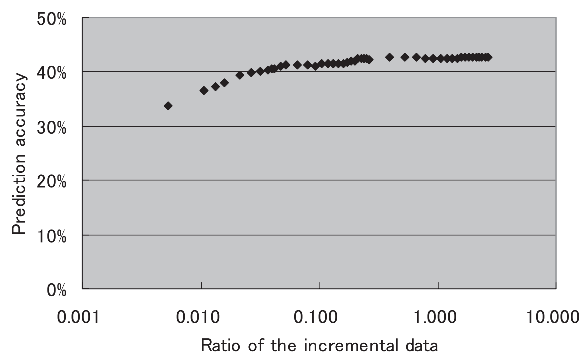

Fig. 10 Prediction accuracy with the ratio of incremental data.

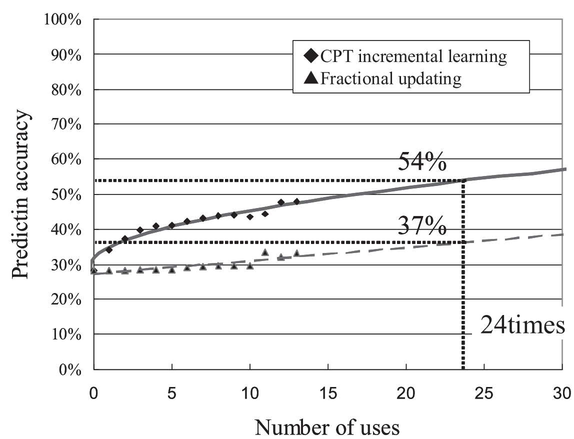

Fig. 11 Prediction accuracy by incremental learning.

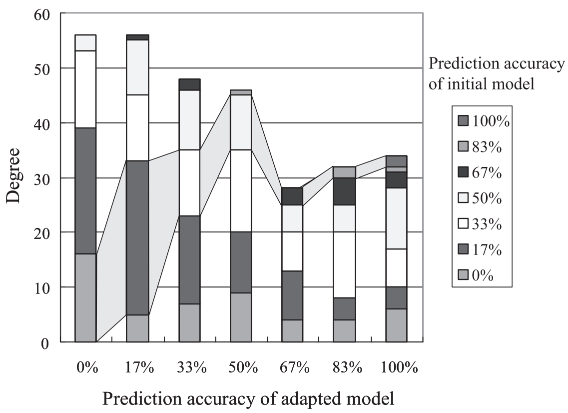

Fig. 12 Effect for each individual subject by incremental learning.

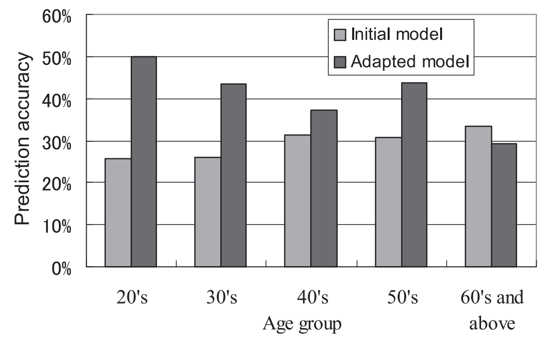

Fig. 13 Change of prediction accuracy of each age group.

the standard model of a subject with the same attributes. Therefore, since the difference was small, the model could adapt even with a small number of samples. In contrast, in the subject group whose results in the initial model were bad (33 % or less), some subjects achieved 100 % with only five samples (the lower half of the rightmost rod). Although preferences of such subjects differed from those of the standard model, the subjects could be considered to have content preferences that did not depend on situations. These results demonstrate that CPT incremental learning is stable for users with high prediction accuracy and has high adaptation performance for users with preferences that have low dependence on the situation.

Figure 12 uses hatching to connect the frequency of the subjects whose prediction accuracy did not change. The area below the hatching indicates subjects whose prediction accuracy improved through adaptation, and the area above indicates the opposite. Fifty percent of the subjects improved; 24 % did not change; and 26 % degraded. Adaptation by CPT incremental learning was applicable more than 74 % of the subjects even with only five samples.

As a result, we found that learning a user's individual preference through CPT incremental learning would be beneficial to many users, even with small numbers of samples, and that incremental learning is a necessary technology for realizing user-adapted mobile systems.

Additionally, we analyzed the kind of subjects whose adaptation improved or degraded. Figure 13 indicates prediction accuracy by age group. In the 20 s and 30 s age group, accuracy improved significantly through adaptation, despite the low prediction accuracy in the initial model, revealing remarkable effectiveness of adaptation. However, in the 60 s and above age group, results were the opposite. While younger subjects tended to deviate from the standard model, their dependence on situations was low. In contrast, older subjects were generally closer to the standard model, and their criteria of restaurant selection varied with situations. We assumed that prediction accuracy dropped because the versatility of situations could not be expressed with the small number of samples used in this evaluation. To investigate these characteristics, we analyzed the distribution of selected restaurants by each generation, using data from the web questionnaire. Within each generation, we evaluated the variation among people with the same attributes and the variation among situations for each person by the entropy of the selected restaurant attributes. Results indicated that for entropy among people, the 60 s and above age group (average entropy over attributes = 0.845) was less than that of the 20 s age group (average entropy over attributes = 0.863). However, for entropy among situations, the result was the opposite (1.014 and 1.087). From these results, we assumed that younger people are individualistic, whereas older people appreciate situations. Our proposed methods clearly indicate these characteristics of each generation. According to this study, the following solutions can be considered for prediction accuracy.

- For the model of younger age groups, we should increase user attributes.
- For the model of older age groups, we should increase situation attributes and training samples.

Dividing the model according to user attributes such as age could effectively improve prediction accuracy.

6. Conclusion

In this paper, we confirmed the effective use of a Bayesian network to realize user-adapted recommendation of content on mobile devices. We proposed the LK method for modeling, which complements incomplete and insufficient samples with knowledge data, and CPT incremental learning for adaptation from a small number of samples. In order to evaluate the proposed methods, we applied them to restaurant recommendations on a car navigation system. Our evaluation results confirmed that our model based on the LK method could be expected to provide better generalization performance than that based on a conventional method. Furthermore, our system would require much less operation than current car navigation systems from the beginning of use. Our evaluation results also indicated that learning a user's individual preference through CPT incremental learning would be beneficial to many users even with a small number of samples, and the method has better accuracy than the conventional method. As a result, we have developed the technology of a system that becomes more adapted to a user the more it is used.

In this paper, we evaluated a system using basic algorithms and models. For further improvement in prediction accuracy, we will consider predicting the decision-making strategy and establishing a segmentalized model in accordance with user characteristics. Moreover, for extension of the scope of this system, we will apply the technology such as inference for variables such as unacquirable variables and latent variables.

References

[1] http://g-book.com
[2] http://www.onstar.com
[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering. of netnews," Proc. 1994 ACM Conf. on Computer Supported Cooperative Work, pp.175–186, 1994.
[4] M. Brunato and R. Battiti, "PILGRIM: A location broker and mobility-aware recommendation system," Proc. IEEE PerCom 2003, pp.265–272, March 2003.
[5] M. Pazzani, J. Muramatsu, and D. Billsus, "SYSKILL & WEBERT: Identifying interesting web sites," Proc. Thirteenth National Conference on Artificial Intelligence, pp.54–61, AAAI Press, Portland, OR, 1996.
[6] S. Long, R. Kooper, G.D. Abowd, and C.G. Atkeson, "Rapid prototyping of mobile context-aware applications: The Cyberguide case study," Proc. 2nd Annual. International Conf. on Mobile Computing and Networking, pp.97–107, 1996.
[7] P. Maes, "Agents that reduce work and information overload," Commun. ACM, vol.37, no.7, pp.30–40, 1994.
[8] K. Bradley, R. Rafter, and B. Smyth, "Case-based user profiling for content personalization," Proc. International Conf. on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2000), pp.62–72, Trento, Italy, 2000.
[9] T. Jennings and H. Higuchi, "A personal news service based on a user model neural network," IEICE Trans. Inf. & Syst., vol.E75-D, no.2, pp.198–209, Feb. 1992.
[10] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
[11] E. Horvitz, P. Koch, R. Sarin, J. Apacible, and M. Subramani, "Bayesphone: Precomputation of context-sensitive policies for inquiry and action in mobile devices," Proc. Tenth International Conference on User Modeling (UM05), pp.251–260, 2005.
[12] H. Prendinger, J. Mori, and M. Ishizuka, "Recognizing, modeling, and responding to users' affective states," Proc. Tenth International Conf. on User Modeling (UM05), pp.60–69, 2005.
[13] C. Ono, M. Kurokawa, Y. Motomura, and H. Asoh, "A context-aware movie preference model using a Bayesian network for recommendation and promotion," User Modeling 2007 LNAI 4511, pp.247–257, 2007.
[14] G.F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," Artif. Intell., vol.42, pp.393–405, 1990.
[15] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from Data," Mach. Learn., vol.9, p.309, 1992.
[16] H. Eriksson, M. Penker, B. Lyons, and D. Fado, Uml 2 Toolkit, Wiley, 2003.
[17] H. Peng and C. Ding, "Structure search and stability enhancement of Bayesian networks," Proc. Third IEEE International Conf. on Data Mining (ICDM'03), pp.621–624, 2003.
[18] N. Friedman and M. Goldszmidt, "Sequential update of Bayesian network structure," Proc. Thirteenth Conf. on UAI, pp.165–174, 1997.
[19] D.J. Spiegelhalter and S.L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," Networks, vol.20, pp.579–605, 1990.
[20] M. Fishbein and I. Ajzen, Belief, attitude, intention and behavior: An introduction to theory and research, Addison-Wesley, 1975.
[21] H. Akaike, "A new look at the statistical model identification," IEEE Trans. Autom. Control, vol.19, no.6, pp.716–723, 1974.
[22] J. Suzuki, "A construction of Bayesian networks from databases based on an MDL principle," Proc. 9th Conf. on Uncertainty in Artificial Intelligence, pp.266–273, 1993.

Hirotoshi Iwasaki received the B.S. and M.S. degrees in Electrical Engineering from Nagoya University in 1988 and 1990, respectively. Ph.D. From 1990 to 2000, he worked for DENSO COOPORATION. Since 2000, he has been on loan to DENSO IT LABORATORY, Inc. His research interests include in-vehicle information systems and intelligent interface, and recommender systems. He is a member of JSAI and IPSJ.

Nobuhiro Mizuno received the B.S. degree in Mathematical Engineering from Nihon University in 1996. From 1996 to 2001 he was with Kawasaki Steel Systems R&D Corporation. Since 2001, he has worked for DENSO IT LABORATORY, Inc. His research interests include the boundary between consciousness and unconsciousness. He is a member of JSAI and IPSJ.

Kousuke Hara received the B.S. and M.S. degrees in Electrical Engineering from Tokyo University of Agriculture and Technology in 2000 and 2002, respectively. From 2002 to 2004 he worked at Yokogawa Electric Corp. Since 2004, he has worked for DENSO IT LABORATORY, Inc. His research interests include scene understanding.

Yoichi Motomura received the B.S. and M.S. degrees in Communications and Systems Engineering from the University of Electro-Communications in 1991 and 1993, respectively. Ph.D. Since 1993, he has worked for the National Institute of Advanced Industrial Science and Technology. He also conducts the venture company, Modellize Inc. His current research works are statistical learning theory, probabilistic inference algorithms on Bayesian networks, and their applications to user modeling and human behavior understanding and prediction from sensory data.