Uncertainty Reasoning and Chance Discovery

Kyung-Joong Kim[1] and Sung-Bae Cho[2]

Department of Computer Science, Yonsei University, 134 Sinchon-dong,
Seodaemun-gu, Seoul 120-749, Korea
[1]kjkim@cs.yonsei.ac.kr
[2]sbcho@cs.yonsei.ac.kr

Summary. Chance discovery is one of the hottest issues and various computational methods are applied to solve the problem. It inherently contains uncertainty because it is based on the human's concept and sometimes hard to define clearly. In this domain, it is very important to model human's uncertain knowledge easily and provide manipulation method for the intervention of humans in the knowledge discovery. Recently, Bayesian network that is a symbolic model with flexible inference capability has gained popularity from various domains. In this paper, we present the possibility of Bayesian network for chance discovery in terms of uncertainty reasoning, and show some of the applications such as detecting chances from system, application and concept levels.

1 Introduction

Chance discovery is to recognize a chance which is a very rare event, but with significant impact on decision making or future change [18]. It gives not only an awareness of chances but also an explanation about chances. This approach has been applied to various applications domains such as predicting earthquake, discovering new topics from WWW, and agent communication [1, 3]. First of all, let us discuss about the relationships between chance discovery and uncertainty reasoning.

Recently, uncertainty handling for chance discovery has gained interest with many different approaches such as neuro-fuzzy [20], probabilistic logic [19], qualitative methods [18], and rough sets [17]. Because chance is a rare event, computers have a difficulty to deal with it. In fact, the cooperation of human knowledge and computational algorithm is important to detect the chance. Also, usually chance is considered as something connected with randomness and measured in probability [18]. Therefore, representing human's uncertain knowledge and statistical prior knowledge in a formal inference/prediction model is one of critical problems to deal with uncertainty for chance discovery. There are various sources of uncertainty in chance discovery.

K.-J. Kim and S.-B. Cho: *Uncertainty Reasoning and Chance Discovery*, Studies in Computational Intelligence (SCI) **30**, 83–100 (2006)
www.springerlink.com

- Human: Human is one important component for the success of chance discovery and he has inherent uncertainty. It is difficult to guarantee that his interpretation of provided evidence from computer will be exact. Also his concept about chance is not clear to capture because of vagueness and uncertainty.
- Data: Data could have contradictions. Most of data tell general trends but some of them indicate very rare cases that are significantly opposite compared to the trends. It can be interpreted as a uncertainty because it could has many different results given the similar situation because of hidden effect or inter-relationships among variables. In traditional data mining, data deviated significantly from general trends are considered as a noise but in chance discovery, they are considered as valuable sources for chance and ignoring them must be avoided.
- Chance: Sometimes, chances are defined deterministically but it is not true in many real world situations. Given the same situations, it is unclear that the chance would come true. It can be defined as a concept of probability or degree. Also, the definition of chance has uncertainty. Its boundary that defines chance is not clear and some fuzzy or rough method is needed.

There are various methods to deal with uncertainty such as fuzzy logic [12], rough sets [13], probabilistic models [14], and hybrid of them. They are all very important method to represent uncertainty in chance discovery and they are not competitive but complementary. In this paper, we would like to introduce a variety of criteria for uncertainty handling models in chance discovery and Bayesian networks, one of the representative models in probabilistic models, in details. Bayesian networks are frequently used to model complex cognitive functions of human and show successful results in various applications [21]. The most important decision criteria for uncertainty handling model of chance discovery are easy construction of model with learning algorithm, flexible inference capability with missing variables in two different directions (forward and backward), interpretability and easy manipulation of model for the interaction of humans, and sound mathematical foundations.

- Learning: Basically, chance discovery is a kind of intercommunication process between human and computer because chance is very difficult to identify. Computer supports human by providing evidences about chance from massive dataset using mining algorithm. Its purpose is not to discover general trends of data set but to notify unknown or novel important factors that result in great phenomenon. Though human's role in this procedure is critical, it is natural that doing such task without computer's help is impossible because of huge size of dataset. Therefore, the capability of learning model from the dataset is one of criteria for uncertainty handling and its nature is a bit different from the traditional one in the classification or prediction domain. Because it discovers very rare significant event, showing only good classification or prediction accuracy is not always desirable. Instead of increasing classification accuracy for one effect, it is necessary to model accurate global joint probability distribution that embodied rare events. It is well known that describing an exact joint distribution probability given several variables is not possible due to the huge storage space requirement with a number of parameters. Bayesian network is an acyclic directed graphical model that represents joint distribution probability of random variables. By ignoring irrelevant conditional dependencies among variables, it approximates true joint probability distribution with a small number of parameters. Learning Bayesian networks are well studied for several years and one can easily construct Bayesian networks for this massive dataset.

- Flexible inference: One requirement for chance discovery model is an ability that can deal with novel or unknown situations. Rules are one of the easiest and good choices for modeling trends in massive dataset but it has a critical weakness. In the perspective of uncertainty handling, rule is quite deterministic and cannot deal with contradictions that are critical to model chance discovery. Of course, it is possible to do such thing in the framework of rules with some additional effort (defining weights for each rule or maintaining separate rule set for chance discovery) but it is not competitive compared to other methods that have such capability inherently. Also, rule has a difficulty to model unknown situation (conditions). If rule-set cannot cover all areas of condition space, it cannot provide results for previously unknown conditions. Defining rules manually is very difficult job given a number of variables and it is not surprising that some situations are not defined because of limited information and designer's mistake. Furthermore, accurate description of condition space results in a number of rules. Because Bayesian networks represent joint distribution probability of random variables, it is not necessary to describe conditions for specific results and it can provide evidences for any combinations of variables.
- Missing variables: Chance discovery is related to discover hidden cause-effect relationships that lead to great success or risk. If it has a form of simple cause-effect, rule-based approach can perform well but it is not true. In many cases, some causes might generate great effect and its relationship is difficult to describe in a simple cause-effect form. Its relationship is not direct but indirect one. In the route from cause and effect, there are a number of paths and its length is larger than two. The effect of causes is propagated through indirect links of a number of random variables and finally it results in change of effect variables. In the worse case, some relevant variables can be missing. If there is only a set of condition-results rules, it has a difficulty to provide results given some missing variables. Because Bayesian networks are based on the network structure and probability propagation algorithm to infer the probability of effect, such situation can be easily solved. In the case of missing, it can generate results robustly because the inference algorithm can deal with such situation naturally.

- Bi-directional inference: If there is very successful case or risk, it is interesting to infer the conditions that make it possible. We call it as backward inference. That means uncertainty handling methods require a flexibility of evidence-query setting. Sometimes evidences can be query and the opposite side must be possible. Fuzzy rules that are very useful tool to deal with human's perception in the form of if-then rules need to define additional rules to deal with such cases. If there is enough knowledge about the domain in the form of if-then rules, fuzzy rules can be very good choice but doing backward inference given known results is not easy to do. In the case of Bayesian network, any node (variable) can be query or evidence nodes and there is no restriction. Domain expert can analyze the probability of unobservable evidences given known results (great success, risk and general situation) using backward inference and the knowledge can be used to refine the model to improve the performance of chance discovery.
- Interpretation: Connectionist models such as neural networks for uncertainty handling has a weakness because they are very difficult to understand for human. Domain experts need to analyze the automatically learned model and their inference flow to guess the success or risk of future. For improvement, understanding the model is very important. In this reason, a model like keygraph is preferable because it provides easily understandable visualization of model [1]. Like keygraph, Bayesian networks are based on the graph structure and it is easy to visualize. Each edge of the network represents cause-effect relationships (conceptual relationships) and human expert can understand the whole procedure of inference from cause and effect. One of the successful application areas of Bayesian network is medical area and its reason is that various relationships from doctors knowledge have a form of cause-effect. If a domain has enough cause-effect relationships, complex phenomenon for chance can be easily visualized in the form of network of cause-effect relationships.
- Manipulation: Ohsawa defined the procedure of chance discovery as continuous interactions of human and computers [1]. That means continuous revision of the model is also needed. After understanding the model, human expert could modify the model to represent his belief about chance. If the modification is very difficult, the interaction of computer and human may not be desirable. Modifying rules is relatively easy because it is based on the simple condition-results. The modification of condition parts immediately results in desirable results. However, modifying one rule need to check occurence of contradiction with all other rules. On the other hand, modifying Bayesian network can be thought as an easy process but sometimes it can be difficult because it is based on indirect relationships. Because their edge is based on cause-effect relationship, modification is just procedure of adding/deleting edges with parameter tuning. In case of Bayesian network, there is no guarantee that modification of specific edge can results in desirable one. Because it models joint distribution probability, change of small modification can change the distribution and it can

make unexpected results. Basically, modifying the model is not difficult but tuning the joint probability distribution as you want is a bit difficult. Recommendation for this situation is re-training the parameters of the changed model from accumulated data. However, it is one integrated model compared to rule sets, contradiction check is not necessary.
- Mathematical foundation: Uncertainty handling methods have sound mathematical definition. Their original research was based on mathematical theory and after several years it was applied to many real-world applications. Because of the relatively short history, there is little research for the mathematical foundation of uncertainty handling for chance discovery. It might be better to adopt previously well-defined method in other domains and refine the model for chance discovery. The mathematical foundation of Bayesian networks are from probability theory and its sound mathematical formulation allows researchers focus on more advanced topics.

Bayesian networks use probabilities and assume that it cannot know everything. That allows them to capture subtle behaviors that would require thousands of strict rules [4, 5]. Horvitz et al. propose the construction of Bayesian networks that provide inferences about the probability that subjects will consider events to be memory landmarks based on the intuition that rare contexts might be more memorable than common ones [16].

In this paper, we show some possibilities of the Bayesian networks to detect various chances from middleware reconfiguration (low level), object detection (middle level), and common-sense modeling (high level). In the low level, the knowledge about the rare event is not easily defined by human because of the complexity; it is easy to learn the detection model from data. In the middle level, there are a number of data and also limited knowledge about intermediate concepts. It might be better to model the network with hybrid of learning and modeling by experts. In the high level, there are few data for conceptual situation, and it might be better to model such cases only by human experts.

In middleware, a system fault is relatively rare event and detecting it from various uncertain (due to distributed computing) information sources is a quite challenging. Detecting the critical problem and acting properly based on the prediction is one of the significant decision problems for adaptive middleware. In the framework of component-based middleware, the decision of reconfiguration of components is inferred based on the probabilities from automatically learned Bayesian networks.

Detection of rare objects from visual scene for service robot is one of the interesting applications of chance discovery. Some critical objects such as dangerous toxic, broken cups and unknown things are very important for security, surveillance, and elderly care. In this paper, activity-object Bayesian network is presented to deal with occluded objects and reduce modeling cost [7]. Finally, modeling commonsense for context-awareness of service robot based on Bayesian network is illustrated with hierarchical organization of a number of models. By modeling unexpected situations using probabilistic models, it can detect chances that are crucial to make natural interactions between human and computer [6].

2 Bayesian Network for Chance Discovery

Many researchers think of the Bayesian network as a useful tool for handling uncertainty. Given partially observed variables, BN can give a concise specification of any full joint probability distribution. The design variables or the environmental parameters are subject to perturbations or deterministic changes. It is very common that a system to be optimized is expected to perform satisfactorily even when the design variables or the environmental parameters change within a certain range. If some of variables become unobservable suddenly, conventional optimized solutions might be out of orders. However, probabilistic reasoning tool can deal with such situations naturally without additional effort such as re-optimization. Moreover, BN provides the system designer with cost-effective method of a structural refinement because it is based on a symbolic representation which is easily understandable.

Bayesian probabilistic inference is one of the famous models for inference and representation of the environment with insufficient information. The node of Bayesian network represents random variable, while the arc represents the dependency between variables [8, 9]. In order to infer the network, the structure must be designed and the probability distribution must be specified. Usually the structure is designed by expert while the probability distribution is calculated by expert or collected data from the domain. By observing evidence, the probability of each node is computed by Bayesian inference algorithm based on the conditional probability table and independence assumption.

We use $< B, \Theta_B >$ to denote a Bayesian network with a structure B and probability parameters Θ_B. $P < B, \Theta_B >$ denotes the joint probability distribution of all the variables of this network. A Bayesian network is a directed acyclic graph $B = (V, E)$, where the set of nodes $V = \{x_1, x_2, ..., x_n\}$ represents the domain variables and E, a set of arcs, represents the direct dependency among the variables. For each variable $x_i \in V$, conditional probability distribution is $P(x_i|Pa(x_i))$, where $Pa(x_i)$ represents the parent set of the variable x_i.

$$P < B, \Theta_B >= P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i|Pa(x_i)) \tag{1}$$

Recently, some researchers attempt to deal with uncertainty in Bayesian network learning and modeling. Kim et al. adopt expandable Bayesian network (EBN) for computing 3D object descriptions from images [10]. One challenge in the problem is that the number of the evidence features varies at runtime

Table 1. Joint probability distribution for V(T=True, F=False)

x_1	x_2	x_3	$P(x_1, x_2, x_3)$
T	T	T	P_1
T	T	F	P_2
...	...	...	...
F	F	F	$1 - \sum P_i$

because the number of images being used is not fixed and some modalities may not always be available. It uses repeatable and hidden nodes to deal with the uncertainty. Lam proposes a new approach to refining Bayesian network structures from partially specified data [11].

2.1 Basic Definition

If there are n random variables and they have binary states, the total number of parameters for a joint probability distribution of n variables is $2^n - 1$. For example, V has three random variables x_1,x_2, and x_3. The joint probability of the variables is $P(x_1, x_2, x_3)$. To describe the probability distribution, 7 parameters are needed. Table 1 summarizes the parameters and values of the variables.

If all the parameters for the joint probability distribution are determined, any kind of queries can be calculated using the distribution. For example, $P(x_1 = T)$ can be calculated by the sum of $P_1 + P_2 + P_3 + P_4$. If there is prior knowledge about the domain, the posterior probability of some variables can be calculated using the Bayes rule.

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)} = \frac{P(x_2|x_1)P(x_1)}{P(x_2)} \tag{2}$$

If the values of some variables are known, it is possible to infer the probability of the states of the unknown variables. For example, the value of variable x_i is known as True but the values of x_2 and x_3 are unknown. The probability of x_2=True and x_3=True given A=True is defined as $P(x_2 = T, x_3 = T|x_1 = T)$. Using the Bayes rule, this can be calculated as follows.

$$P(x_2 = T, x_3 = T|x_1 = T) = \frac{P(x_2 = T, x_3 = T, x_1 = T)}{P(x_1 = T)} = \frac{P_1}{P_1 + P_2 + P_3 + P_4} \tag{3}$$

This means that the probability of the unknown variables can be calculated (given some unobserved variables) without additional effort. This flexibility allows robustness of inference against sudden input missing.

In the formula, the variable x_1 is called an evidence variable and x_2 is a query variable. If the value of x_2 is known and the query variable is x_1, the probability of the query is as follows.

$$P(x_1 = T|x_2 = T) = \frac{P(x_1 = T, x_2 = T)}{P(x_2 = T)} = \frac{P_1 + P_2}{P_1 + P_2 + P_5 + P_6} \quad (4)$$

In fact, any variables can be called query nodes, and the probability of these query nodes can be calculated. For example, suppose that there are 3 relevant input variables and 1 chance variable in a chance discovery system. Each input variable, respectively, is denoted by x_1,x_2, and x_3. If the system observes x_1=True, x_2=True but no information x_3, the probability of chance $P(x_4)$ can be calculated as follows.

$$P(x_4 = T|x_1 = T, x_2 = T) \quad (5)$$

If there is no chance, the probability of x_3 can be calculated as follows.

$$P(x_3 = T|x_1 = T, x_2 = T, x_4 = F) \quad (6)$$

This flexibility of inference is very useful in domains with uncertainty. The classification of query and evidence nodes is not clear and some variables have the possibility of sudden loss of information (unobservable). However, there are practical problems for such inference because of the large number of parameters when working with a large number of variables.

Figure 1 shows a simple BN with 5 variables and 4 edges. This means that (x_1, x_3), (x_2, x_3), (x_3, x_4), and (x_3, x_5) are conditionally dependent. $P(x_1, x_3, x_4, x_5)$ is as follows.

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_3)P(x_5|x_3) \quad (7)$$

This calculation needs only 10 parameters instead of $2^5 - 1$. These parameters are $P(x_1 = T), P(x_2 = T), P(x_3 = T|x_1 = T, x_2 = T), P(x_3 = T|x_1 = T, x_2 = F), P(x_3 = T|x_1 = F, x_2 = T), P(x_3 = T|x_1 = F, X_2 = F), P(x_4 = T|x_3 = T), P(x_4 = T|x_3 = F), P(x_5 = T|x_3 = T)$, and $P(x_5 = T|x_3 = F)$. Each child node contains conditional probability parameters. By assuming conditional independence between variables, we can ignore some parameters. The ALARM network, a well-known benchmark BN, has 37 variables and each variable has $2 \sim 4$ states. The number of parameters of the network is 590 instead of 2^{54}.

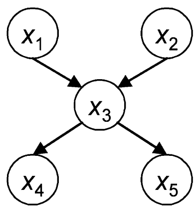

Fig. 1. A simple Bayesian network

2.2 Learning

If there is enough domain knowledge and an expert can summarize this knowledge into a cause-effect form, it can be easily converted into BNs. However, this kind of expert knowledge is only available in popular domains such as medicine and trouble-shooting. Understanding and predicting the behavior of computer systems is very difficult because of the dynamic and complex interactions of modules within the system. To develop appropriate BNs for this specific problem given a little prior knowledge, learning is essential. Learning BNs are composed of two stages: structure and parameter learning. Using a scoring function that returns a numerical value for the appropriateness of the given data in the BNs, search algorithms such as greedy and genetic algorithms attempt to maximize the score.

In order to induce a BN from data, researchers proposed a variety of score metrics based on different assumptions. Yang et al. compared the performance of five score metrics: the uniform prior score metric (UPSM), the conditional uniform prior score metric (CUPSM), the Dirichlet prior score metric (DPSM), the likelihood-equivalence Bayesian Dirichlet score metric (BDe), and the minimum description length (MDL) [22]. They concluded that the tenth-order DPSM is the best score metric. If the Dirichlet distribution is assumed, then the score metric can be written as [23, 24].

From an empty network with no edge, the greedy algorithm repeats the procedure of adding an edge that maximizes a score gain on the current structure and fixes the new structure as a current one until the structure converges. Though the algorithm can get stuck in the local minimum, it can perform well if the number of variables is relatively small. If the number of variables is large, a global search algorithm such as a genetic algorithm is a more appropriate choice. In this domain, we assume that relevant variables are selected by the expert and the learning procedure is conducted by the greedy algorithm. Figure 2 shows the pseudo algorithm of the greedy search.

3 Case Studies: Three Applications

In this section, we would like to present about three applications of Bayesian network for chance discovery. Each application represents one of the three different levels of chances. Low-level chance is likely to subsymbolic (hard to be understood by humans) because it is based on complex interactions of low-level sensors. In this application, we apply Bayesian network to detect rare system fault before it occurs. Based on probability information, users can be prepared for critical system-down by swapping some components (parts) of applications. In middle-level, something human-understandable but based on data-driven induction concepts is main focus of chances. Detection of relevant objects given occlusions is challenging tasks and proper detection of objects can give service robot opportunity of doing right job or avoiding severe risks.

```
/* n : Number of variables */
/* A[i,j] : Score gain when an edge from jth node to ith node is connected */
/* Score(B) : Score of Bayesian network structure B */
/* Score(B,j→i) : Return a score when B has an edge (j→i) */
/* Find_Max(A) : Return an edge (j→i) that has an maximum A[i,j] */
/* Min : minus infinity */
/* Ancestors(x_i) : A set of nodes that have a path from the node to x_i */
/* Decendants(x_i) : A set of nodes that have a path from x_i to the node */
/* Stop(); If all (i,j), A[i,j]≤0 or A[i,j]=Min then True */
FOR i=1 to n { Pa(x_i) = φ; }
FOR i=1 to n { FOR j=1 to n {
     IF i≠j THEN A[i,j]=Score(B,j→i) – Score(B); }}
WHILE(TRUE){
  (i,j)=Find_Max(A);
  IF A[i,j]>0 THEN Pa(x_i) = Pa(x_i) ∪ {x_j};
  A[i,j]=Min;
  FOR a=1 to n { FOR b=1 to n {
   IF x_a ∈ Ancestors(x_j) ∪ {x_j} && x_b ∈ Decendants(x_i) ∪ {x_i}
   THEN A[a,b]=Min; }}
  FOR k=1 to n {IF A [i,k]>Min THEN A[i,k]=Score(B, k→i) – Score(B); }
  IF Stop()==True THEN break;
}
```

Fig. 2. Pseudo code of greedy search for Bayesian network

In this application, we present template-based modeling method of Bayesian network. By reducing the design space of Bayesian network, the model can be easily designed by human expert and machine learning algorithm. Finally, high-level chances are detected based on human-level concepts. It is based on the common sense of humans and chances can be modeled by human as a prior knowledge. Exceptional cases of human knowledge are encoded in the Bayesian networks by the expert.

3.1 Case 1: Low-level

BNs are used to detect future system faults for the decision module that selects appropriate components in component-based middleware. Some information from remote servers through networks (such as availability of specific components, system resource status, and network accessibility) might be uncertain if there are unexpected delays caused by network congestion. Given information from system, the middleware automatically reconfigure

Table 2. Accuracy of the learned Bayesian network on the test data (Time interval = 3 seconds)

	CPU LOAD>8.6	CPU LOAD<8.6	Accuracy
CPU LOAD>8.6	173	11	94.02%
CPU LOAD<8.6	11	953	98.85%
Total	184	964	96.44%

the component-based applications by adding, deleting, and replacing components (building blocks). The probabilistic information about the system's future status can be visualized or utilized by human expert or other decision theory to decide appropriate actions. The Bayesian network provides the probability of future system faults given current system resource and application information.

The basic idea of reconfiguration is as follows. For each functional group, there are a number of redundant components with different properties. In the reconfiguration procedure, the functional graph of the application will not change but each component of the function is replaced with an appropriate component with the same function. The decision of selecting an appropriate one from the functional group is based on the inferred results from the BNs. Evidence is inputted into the BNs with uncertainty and this provides belief about the query nodes with probability. Finally, component-swapping rules are used and their conditional parts contain variables that are inferred from the BNs. Figure 2 summarizes the procedure.

Figure 6 shows the structure of the automatically learned BNs. Data are collected from users for 90 minutes (30 minutes for learning, 60 minutes for test) on linux server. If the CPU load is larger than predefined threshold, it is recognized as system fault (At the threshold, some multimedia player works wrong). Using current CPU resource and application usage information, the BN estimates the probability of CPU overload. The percentage of overload is relatively low and it is estimated before it occurs. Table 3 shows the accuracy of the learned models on test data. It shows 96.44% accuracy. Especially, its accuracy on the positive samples (true) is 94.02%.

3.2 Case 2: Middle-level

In indoor environments, vision-based service robot requires to account for undetected objects that are too small or occluded by others (can be considered as a chance for relevant acvity recognition). The Bayesian network designed to model the object relationship based on activity is called an activity-object Bayesian network. For each activity, related objects are grouped and their relationships are encoded in Bayesian networks which are used to estimate the probability of object's existence given various obstacles (occlusion). In this section, a template-based approach for building Bayesian networks is proposed.

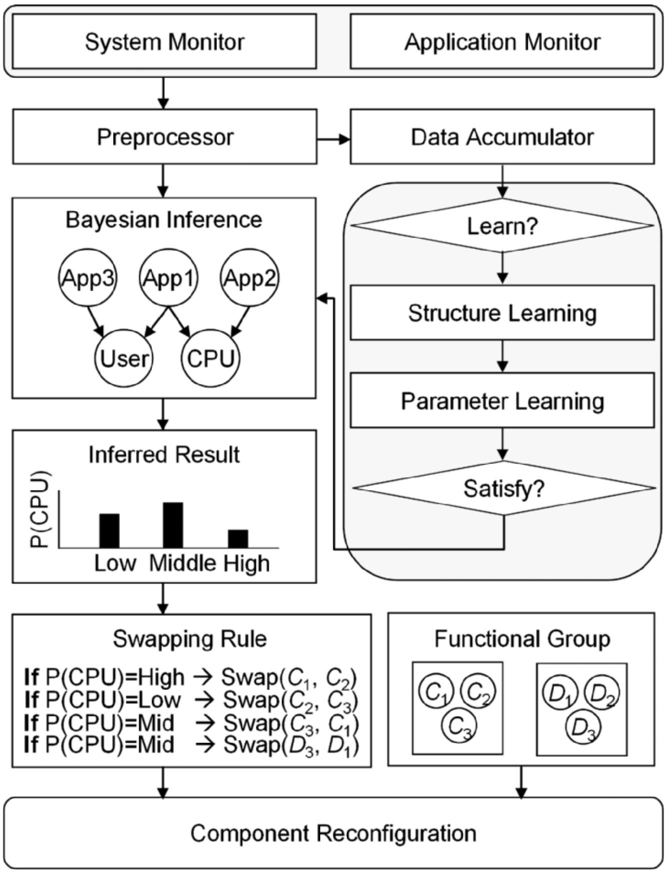

Fig. 3. Component reconfiguration using BNs

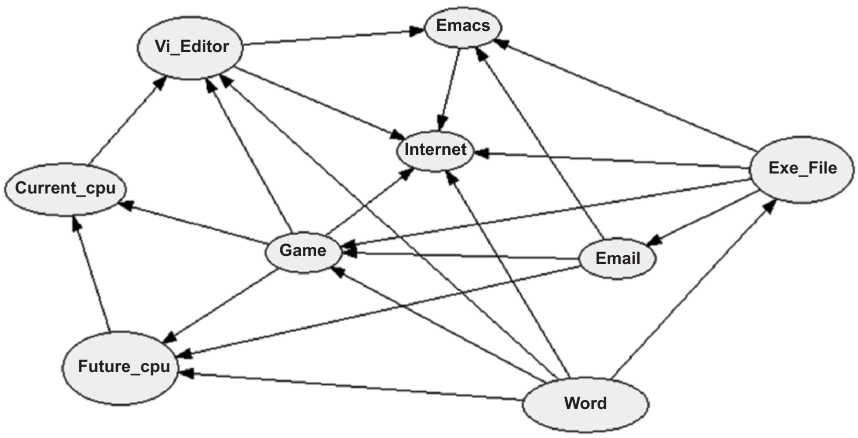

Fig. 4. Bayesian networks learned from data

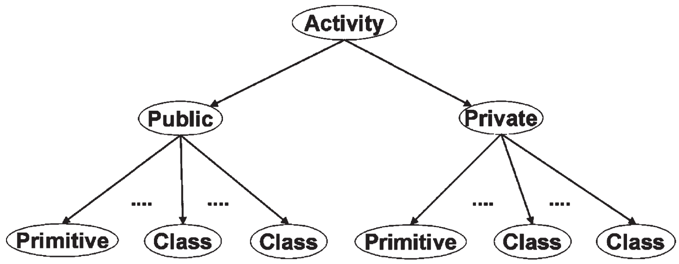

Fig. 5. Basic structure of the activity-object Bayesian network

Classification	Contents
Places	Lecture room, Meeting room, Seminar room, Computer room, Rest room , Prof. office, Admin. office, Guard office, Lab., Stair, Corridor, Elevator, Toilet
Objects	Table, Chair, Round table, Sofa, Cushion, Lectern, Cabinet, Bookcase, Garbage can, Sink, Seat toilet, Wall clock, Air conditioner, Telephone, Computer, Mouse, LCD monitor, Keyboard, Beam Projector, Projection Screen, Audio, Speaker, Microphone, Board, Partition, Curtain, Water bucket, Door, Window, Vending machine, Beverage, Book, Key

Fig. 6. Service environment

The activity-object Bayesian network has a tree structure which is composed of four kinds of basic nodes: an activity node (A), a class node (C), a primitive node (O), and a virtual node (V). The activity nodes are used as root nodes, and they represent the relationship between the objects. Class nodes are used to show more specific relationships between the objects which are known as the building blocks of the Bayesian network. Primitive nodes represent observed or target objects for detection. Virtual nodes are used for adjusting the influence between the objects. We obtain the object relationships by using these kinds of nodes.

Two more nodes are also used as class nodes: public class nodes and private class nodes. Public class nodes are general and can be reused commonly with only slight adjustments in other activity-object Bayesian networks. Private class nodes have specific class nodes related to activity. The object relationship within private nodes is discovered by experts on a case-by-case basis. Public class nodes show weaker relationships to activity, because they are widely used in many cases and include only a small amount of information. The basic structure of the activity-object Bayesian network is shown in Figure 8.

In our experiments, a robot performed services in a university environment. We designed activity-object Bayesian networks which included 15 places and 29 objects (summarized in Figure 9).

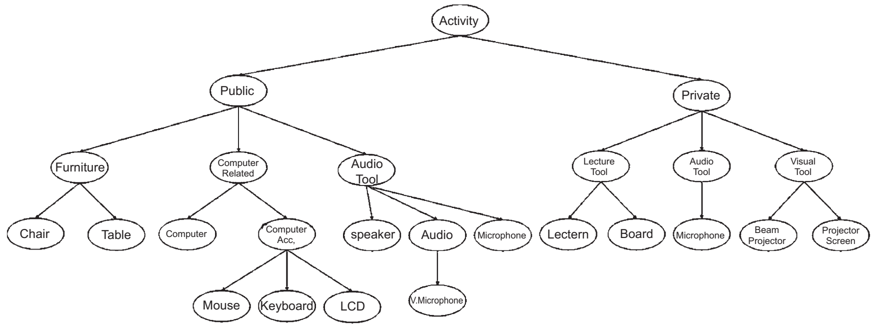

Fig. 7. Presentation activity-object Bayesian network

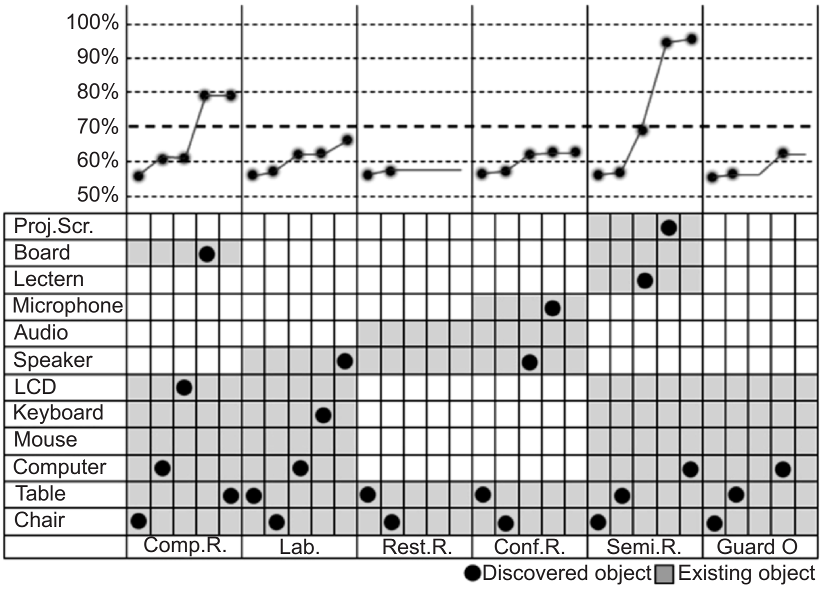

Fig. 8. The changes of probability of the beam-projector

Experiments were carried out to verify the performance of activity-object Bayesian networks in six different places (Computer room, Laboratory, Rest room, Conference room, Seminar room and Guard office). We used two activity-object Bayesian networks: the presentation activity-object Bayesian network was used for finding the beam-projector 11. We assumed that the service robot would move from place to place and randomly detect objects. We recorded the values and hit rates to predict the probability of target objects being present in each place. The target objects refer to the beam-projector.

Figure 14 shows the changes of probability of the beam-projector in the presentation activity-object Bayesian network.

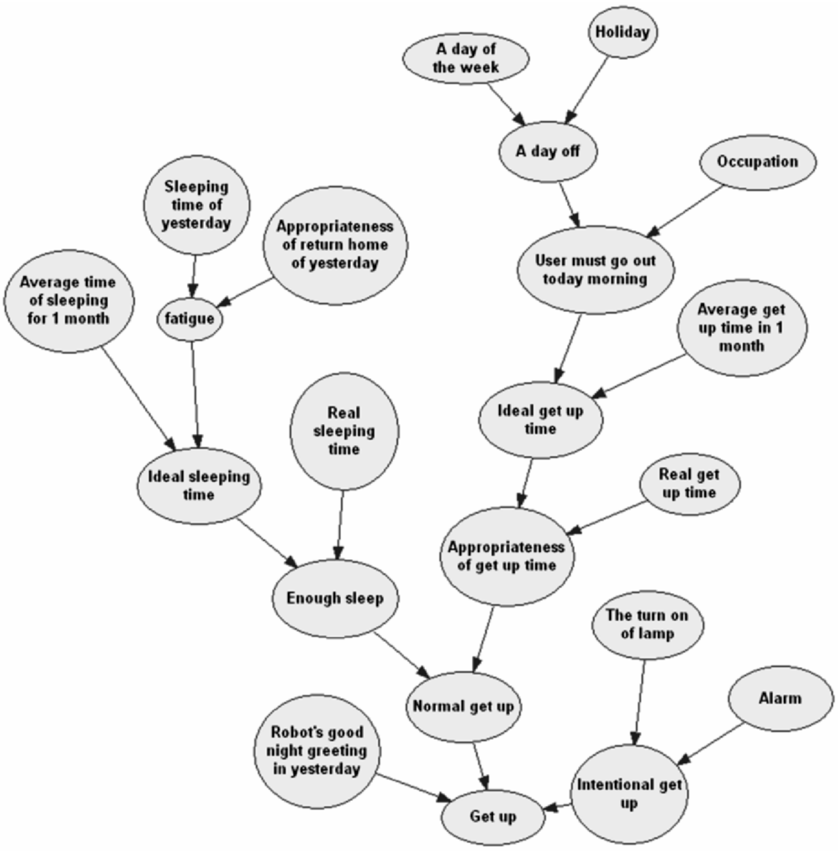

Fig. 9. Morning Bayesian networks

The probabilities of the beam-projector being present in each place were observed under a threshold of 70% until the robot was able to find five objects. The prediction seemed reasonable except in one case. Although the robot predicted that a beam projector existed in both the computer room and the seminar room, in fact, the projector did not exist in the computer room. This denoted that a false-positive error is likely to occur in similar environments that contain many objects related to a beam-projector. Thus, it is important to determine the value of the threshold and the number of times to find objects for each performance.

3.3 Case 3: High-level

Service robot needs to understand common sense of human and exceptional chances that are relevant for human-robot interaction. It is not easy to learn such knowledge from bottom-up manner. We have implemented context-aware system based on uncertain knowledge inference for ubiquitous robot which can access numerous sensors, user profile and calendar information. The robot navigates the inside of home and provides useful services to the user based on the inferred context. For example, they are music recommendation, greeting, and surveillance services. Because the services are very sensitive to the context of the user, it is very important to infer correct context from low-level information.

The robot has many sensors such as humidity, temperature, audio, light, and magnetic sensors. Using RFID tag, the location of the user and objects can be easily determined. Furthermore, the sensor provides the identification of objects (the name of person). If robot recognizes the person using RFID tag, personal information (schedule, name, occupation, income, outcome, relationship with other people, favorites) can be accessible directly from the user profile.

Especially, context-aware system for greeting services is developed using Bayesian networks. The selection of greeting is highly related to the context of the situation. There are four different types of greetings in the morning. Small icons on the top of each situation describe basic information from raw sources such as calendar, time, conversation, light, door, calendar, and temperature. Given some information, Bayesian networks infer the context of each situation. It is not necessarily that all the information is observable and it allows some missing information.

There are 19 different situations for greeting services. We have defined heuristic rules to choose some Bayesian networks. For example, if the user is on the door, Door greeting Bayesian network is selected. In this scenario, the most important situations are morning, night, unexpected and door. The selection of other Bayesian networks is determined based on the inferred results from the major four networks. The average number of nodes and edges for 19 Bayesian networks are 13 and 14, respectively. Because two nodes share one edge, the number of edge is relatively small. Figure 9 shows morning Bayesian network. The robot user the Bayesian networks to guess users current goal. For example, MorningBN is used to guess users getting up. Though user goes out from the bed room, there is possibility that it is not get up. User can get up to go toilet or drink water for a minute. In this case, the robot must provide appropriate greetings. By modeling unusual greetings, the robot can interact naturally with humans. For example, if user returns home very lately at night and there are some friends who visit his house, we can guess that they will drink some alcohol with high probability. If it is very special day such as birthday and event day, the probability will rise up. But if user has very important meeting in the early morning, the probability of drinking alcohol will decrease.

Figure 10 shows the organization of multiple Bayesian networks. Based on the inferred results from the major four Bayesian networks (Morning, Night, Door, Unexpected), other 14 Bayesian networks are selected.

IF robot sees user go out from bed room && time is morning
THEN "Morning" BN
IF robot sees user go to bed room && time is night
THEN "Night" BN
IF user is in the door **THEN** "Door" BN
IF "Get up" is false from "Morning" BN
THEN "Toilet," "Close/Open Window", "Drinking water" BNs
IF "Get up" is true from "Morning" BN
THEN "Reading a book," "Exercise" BNs
IF "Sleeping" is false from "Night" BN
THEN "Studying" BNs
IF "Coming in/out" is false from "Door" BN
THEN "Cleaning," "Cart Away Garbage," "Check mailbox,"
"Going to near store," "Going to car,"
"Check milk or newspaper" BNs

Fig. 10. Rules for selecting Bayesian networks

4 Conclusions

Chance discovery in the uncertain domains will be faced with various challenges such as missing information, unclear definition of chance, and degree of relevance problem. To deal with such challenges naturally, well-defined probabilistic models can be easily used to detect novel chances from the information of sensor, user profile, and user's feedback. Case studies on low-level, middle-level, and high-level domains show that the Bayesian network-based approach might be promising, but there are still a lot of things remained to investigate the relationships between the two fields to exploit the uncertainty reasoning for chance discovery.

References

1. Ohsawa Y, McBurney P (2003) Chance Discovery. Springer, Berlin Heidelberg New York
2. Ohsawa Y (2002) Chance discoveries for making decisions in complex real world. New Generation Computing 20(2): 143-164
3. Ohsawa Y, Yachida M (1999) Discover risky active faults by indexing an earthquake sequence. Proceedings of the Second International Conference on Discovery Science 208-219
4. Onisko A, Lucas P J F, Druzdzel M J (2001) Comparison of rule-based and Bayesian network approaches in medical diagnostic systems. Lecture Notes in Computer Science 2101: 283-292

5. Gibbs W W (2004) Considerate computing. Scientific American January 54-61
6. Kim K J, Cho S B (2005) Construction of Bayesian networks for context-aware service robot. The 2nd International Conference on Ubiquituos Robots and Ambient Intelligence 69
7. Song Y S, Cho S B, Suh I H (2005) Activity-oriented Bayesian networks for detecting occluded objects in uncertain indoor environment. Lecture Notes in Artificial Intelligence 3683: 937-944
8. Charniak E (1991) Bayesian network without tears. AI Magazine 12(4): 50-63
9. Stephenson T (2000) An introduction to Bayesian network theory and usage IDLAP-PR00-03
10. Kim Z, Nevatia R (20003) Expandable Bayesian networks for 3D object description from multiple views and multiple mode inputs. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(6): 769-774
11. Lam W (1998) Bayesian network refinement via machine learning approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3): 240-251
12. Zadeh L A (1965) Fuzzy sets. Information and Control 8:338-353
13. Pawlak Z (1991) Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, The Netherlands
14. Pearl J (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann
15. Engestrom Y (1999) Activity theory and individual and social transformation. Perspectives on Activity Theory, Cambridge University Press
16. Horvitz E, Dumais S, Koch P (2004) Learning predictive models of memory landmarks. CogSci 2004: 26th Annual Meeting of the Cognitive Science Society
17. 1st Annual Workshop on Rougth Sets and Chance Discovery (2005) http://www.jcis.org/pages/subconference/rscd/rscd.aspx
18. Chen X (2005) Cognitive relevance and chance discovery. Readings in Chance Discovery (Abe A, Ohsawa Y eds.)
19. Ogata N (2005) Towards dynamic probabilistic logic: A survey and a proposal. Readings in Chance Discovery (Abe A, Ohsawa Y eds.)
20. Tung W L, Quek C, Cheng P Y K (2005) A neuro-fuzzy perspective to chance discovery and its application to the study of banking failures. Readings in Chance Discovery (Abe A, Ohsawa Y eds.)
21. Horvitz E, Kadie C M, Paek T, Hovel D (2003) Models of attention in computing and communications: From principles to applications. Communications of the ACM 46(3):52-59
22. Yang S, Chang K C (2002) Comparison of score metrics for Bayesian network learning. IEEE Transactions on Systems, Man and Cybernetics-Part A 32(3): 419-428
23. Cooper G F, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Machine Learning 7: 299-347
24. Heckerman D, Geiger D, Chickering D M (1995) Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20(3):197-243