Expert Systems with Applications 39 (2012) 4908–4914

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Context-prediction performance by a dynamic Bayesian network: Emphasis on location prediction in ubiquitous decision support environment ☆

Sunyoung Lee [a], Kun Chang Lee [a,b,*]

[a] Department of Interaction Science, Sungkyunkwan University, Seoul 110-745, Republic of Korea
[b] SKK Business School, Sungkyunkwan University, Seoul 110-745, Republic of Korea

ARTICLE INFO

Keywords:
Dynamic Bayesian networks
Context prediction
Ubiquitous computing
Ubiquitous decision support system
Bayesian network
Naïve Bayesian network
Tree augmented naïve Bayesian network

ABSTRACT

Ubiquitous decision support systems require more intelligent mechanism in which more timely and accurate decision support is available. However, conventional context-aware systems, which have been popular in the ubiquitous decision support systems field, cannot provide such agile and proactive decision support. To fill this research void, this paper proposes a new concept of context prediction mechanism by which the ubiquitous decision support devices are able to predict users' future contexts in advance, and provide more timely and proactive decision support that users would be satisfied much more. Especially, location prediction is useful because ubiquitous decision support systems could dynamically adapt their decision support contents for a user based on a user's future location. In this sense, as an alternative for the inference engine mechanism to be used in the ubiquitous decision support systems capable of context-prediction, we propose an inductive approach to recognizing a user's location by learning a dynamic Bayesian network model. The dynamic Bayesian network model has been evaluated with a set of contextual data from undergraduate students. The evaluation result suggests that a dynamic Bayesian network model offers significant predictive power in the location prediction. Besides, we found that the dynamic Bayesian network model has a great potential for the future types of ubiquitous decision support systems.

1. Introduction

As the ubiquity becomes perceived by users more easily in their daily lives, context-aware systems have received a lot of attention from both practitioners and researchers. World-wild excitement about smart phones is also representing people's passion about the context-aware devices. In this respect, context-awareness has been the subject of the growing attention in the area of ubiquitous computing over the years due to its usefulness for several application domains (Hong, Suh, & Kim, 2008). When computer systems are aware of the context in which they are used and are able to adapt to changes in context, they can engage in more efficient interaction with users.

Context awareness is concerned with enabling ubiquitous computing devices to be aware of changes in the environment, and to intelligently adapt themselves to provide more meaningful and timely decision support for decision-makers (Feng, Teng, & Tan, 2009). However, context-aware systems are limited by the fact that their target is the current context, and that the future context is not predicted by context-aware systems. Therefore, the quality of services provided by the context-aware systems is seriously restricted when future contexts change drastically. To this end, we need to consider the task of context prediction in order to proactively offer high-quality services for users in ubiquitous computing environments.

Context prediction opens a wide variety of possibilities of context-aware computing applications. A context-prediction application may infer the future location of an office owner and redirect incoming calls to the future location. A context-prediction application may also be useful for enhancing the quality of transportation systems. Based on the information about the current location and the future location of a particular user, transportation systems equipped with context prediction technology may be able to assist drivers more effectively by inferring possible preferred routes and by providing customized route suggestions for drivers, as well as warning the drivers about possible dangers by predicting their future context. Knowing the current location and current time, together with the user's calendar, could also allow application to have a good idea of the user's current social situation, such as if the user is in a meeting, in class, waiting in the airport, and so on.

☆ This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea as a result of UCN's subproject 11C3-T2-20S.

* Corresponding author at: Department of Interaction Science, Sungkyunkwan University, Seoul 110-745, Republic of Korea. Tel.: +82 2 7600505; fax: +82 2 7600440.

E-mail address: kunchanglee@gmail.com (K.C. Lee).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.10.026

The list of applications listed here is limited and we believe that there is a great potential for context prediction to be used in a variety of ubiquitous computing applications. Especially, it becomes clear how much users would benefit from the ubiquitous decision support systems equipped with the context-prediction mechanism. As an alternative for the inference engine to be used in the ubiquitous decision support systems capable of providing context-prediction function, this paper proposes a dynamic Bayesian network (DBN) approach to location prediction for ubiquitous computing environments. DBN is an important technique because of its ability to represent the temporal properties of user context information. In fact, it is obvious that a user's current locations are influenced by their previous locations, and particular locations afford particular types of actions. Therefore, we adopted a DBN approach for recognizing the locations of users.

This paper is structured as follows. Section 2 discusses context prediction and various context prediction techniques in ubiquitous computing environments. The modeling techniques used to predict a user' locations are described in Section 3. The results of the experiment are presented and discussed in Section 4, followed by concluding remarks and directions for future work in Section 5.

2. Background

2.1. Context prediction

Context prediction focuses on inferring users' context based on analyzing the observed context history that users have shown so far. The observed context history is a series of context information showing how users are moving around in a certain ubiquitous computing environment. The context information is supplied by various types of sensors such as GPS, RFID, and a variety of wireless devices. These sensors may provide the context information about users' locations, users' actions, or the changes in a physical environment of the users. The purpose of context prediction is to predict the subsequent context that users will likely to enter (if the contexts are locations or situations) or perform (if the contexts are actions) based on a history of contexts which are compiled through various sensors. For ubiquitous computing environments, the ability to accurately predict a user's contexts would make it possible to provide context-aware services that are more natural and customized to people's needs. Accurately recognizing a user's contexts could provide more effective and personalized advices to a user, particularly in ubiquitous decision support systems.

2.2. Context prediction techniques

Several context prediction techniques have been proposed in the literature such as Bayesian networks (Hwang & Cho, 2009; Petzold, Pietzowski, Bagci, Trumler, & Ungerer, 2005), Markov models (Rashidi, Cook, Holder, & Schmitter-Edecombe, in press; Singla, Cook, & Schmitter-Edgecombe, 2010), Topic models (Huýnh, Fritz, & Schiele, 2008; Kim, Helal, & Cook, 2010) and neural network approaches (Petzold et al., 2005). Examples of context prediction are location prediction (Anagnostopoulos, Anagnostopoulos, Hadjiefthymiades, Kyriakakos, & Kalousis, 2009; Laasonen, Raento, & Toivonen, 2004; Petzold et al., 2005), movement prediction (Perl, 2004), action prediction (Brdiczka, Reignier, & Crowley, 2007; Davison & Hirsh, 1998; Singla et al., 2010), daily routine prediction (Huýnh et al., 2008; Kim et al., 2010).

Petzold et al. (2005) investigated Bayesian networks, neural networks, Markov and state predictors to predict the next location of the office owner in an office building. Their system predicted the next location of the office owner and switched over the phone call to the predicted location. Singla et al. (2010) proposed a Hidden Markov Model approach to recognizing activities performed by multiple residents in a single smart home environment. Sensor readings were collected in the smart home environment while participants were performing their activities. Hidden Markov Model was used to determine an activity that most likely corresponds to an observed sequence of sensor readings. Bayesian networks can be used to predict prominent activities of users. For example, Hwang and Cho (2009) proposed a modular Bayesian network model to infer landmarks of users from mobile log data such as GPS log, call log, SMS log, picture log, music-playing log and weather log.

A promising topic model approach to recognizing a user's daily routines has been proposed for ubiquitous computing environments. For example, Huýnh et al. (2008) adopted a topic model to predict a user's daily routine (such as office work, commuting, or lunch routine) from users' activity patterns. To evaluate the topic model, they collected the daily activities of one person over a period of sixteen days. For data collection, the subject wore two sensors in order to record low-level signals such as body movements or body posture. The subject was asked to annotate his activities in detail in order to model the relationship between user activities and low-level signals. In total, 34 activities were recorded in their dataset. Huýnh et al. first identified the user's activity patterns from low-level sensor data by using various classifiers such as Support Vector Machines, Hidden Markov Models, and a Naïve Bayesian network. The resulting user activity patterns that were identified were then given to the topic model as inputs to infer the user's daily routine.

As we have seen so far, there are many examples of context prediction in a variety of application domains. Several strategies can be employed to identify the future location of a user. One such technique is to adapt probabilistic models which predict a user's future location. Section 3 describes an inductive approach to generating location prediction models in ubiquitous computing environments.

3. Inducing location prediction models

Many types of location recognition models can be learned. We investigated probabilistic models such as dynamic Bayesian networks (DBNs), general Bayesian networks (GBNs), tree augmented Naïve Bayesian networks (TANs), and Naïve Bayesian networks (NBNs). Refer to appendix for more details about Bayesian networks that were considered in this study.

3.1. Bayesian models for location prediction

A Bayesian network approach is well suited for generating predictive models in a real-world domain because of its ability to deal with the uncertainty inherent in every facet of human life. Bayesian networks are probabilistic models in the form of directed acyclic graphs (Pearl, 2000). Nodes in Bayesian networks represent variables or propositions (e.g., the occurrence of an event or a feature of an object). Likewise, links represent causal or informational dependencies among variables, and are quantified by the conditional probability of a node, given its parents. If a node does not have parents, it is associated with a prior probability. Since Bayesian networks represent causal or informational dependencies among variables, variables that are not influenced by any other variables but do exert influence on other variables are positioned at the top layer of the network. Similarly, variables that are influenced by some variables and also influence other variables are positioned in the middle layers of a network, while variables that are influenced by some variables but do not influence any other variables are positioned at the bottom layer. In such a representation, it is possible to infer the probability of any combination of variables without having to represent the joint probabilities of the variables.

In general, there are five classes of Bayesian networks. NBN (Witten & Frank, 2005) is the simplest Bayesian network that has one parent node of all other nodes. The parent node is often called a class node. No other links exist in the network. NBN is useful for preliminary predictive model induction due to its naïve independence assumptions. TAN (Witten & Frank, 2005) is formed by adding directional links between attributes in NBN. After removing the class node in a TAN, the attributes should form a tree. GBN (Witten & Frank, 2005) is an unrestricted Bayesian network which treats the class node as an ordinary node. Therefore, the class node can be a child node of some nodes. These Bayesian networks do not provide direct mechanism for representing temporal dependencies among attributes. However most of real-world events are changing over time, which can't be modeled by static Bayesian networks. DBN (Murphy, 2002) provides a systematic way to model the temporal and causal relationships among variables. A DBN is a Bayesian network that represents variables with temporal characteristics, and is composed of a sequence of General Bayesian networks. In a DBN, each Bayesian network represents the state of variables at different times.

3.2. GBN, TAN, and NBN model induction

For model induction, we adopted two different learning techniques: those that are fully automated, and those that require the knowledge provided by a domain expert. We adopt completely automated approach to generate GBN, TAN, and NBN location prediction models. During training phase, the following observable attributes were recorded:

- *User previous actions*: user previous action represents actions that a user took right before a user performs current actions.
- *User current actions*: user current action represents actions that a user currently takes.
- *User locations*: user location represents a location in which a user current action is performed.
- *Routes*: routes represent that a user can take to arrive the current location. The maximum number of routes that a user can take was seven in our experiment. For example, route 1 is the first route that a user can take from the previous location to get to route 2 and route 2 represent the second route that a user can take from route 1 to get to route 3 and so on. Therefore, routes 1–7 comprises a location path from previous location to current location.

Once the dataset is prepared, the user data were loaded into the WEKA machine learning tool (Witten & Frank, 2005), structures and conditional probabilities of GBN, TAN, and NBN were learned, and tenfold cross-validation analyses were run on the resulting models. The entire dataset was used to generate several types of location prediction models. Fig. 1 shows the induced GBN and TAN location recognition models. Because most of users did not record route 6 and 7, nodes for representing route 6 and 7 were missing in the induced GBN.

3.3. DBN model induction

The model induction for DBN proceeds in four phases: (1) identify domain variables; (2) examine dependencies between the domain variables and the manner in which these domain variables change over time; (3) describe how the conditional probability distributions are constructed from the user's action and location data; and (4) procedurally develop the belief update. The domain variables that were used in DBN model included high-level actions (such as talking, walking, and moving) and places (such as a classroom, home, and outdoors) in which users could perform these actions. After identifying domain variables, their dependencies are determined.

The DBN shown in Fig. 2 represents dependencies between actions and locations. Because current actions are affected by preceding actions, there is a directional link from a user action at time $t-1$ to user action at time t. This temporal link models the fact that the domain variables change over time. Further, links from user location nodes to user action nodes represent the fact that a user action is executed in some location.

Once the structure of the DBN is specified, $P(\text{User Action } t|\text{User Action } t-1)$, $P(\text{User Action } t|\text{User Location } t)$, and $P(\text{User Location } t|\text{User Location } t-1)$ are estimated from the dataset in a training phase. Because all possible configurations of variables cannot typically be observed, conditional probabilities of some variables cannot be easily computed from data. For example, if a network has N nodes (variables), and even if each node can have discrete binary values, the total number of possible configurations is 2^N. Therefore, it is often the case that sufficient data are not available to learn the

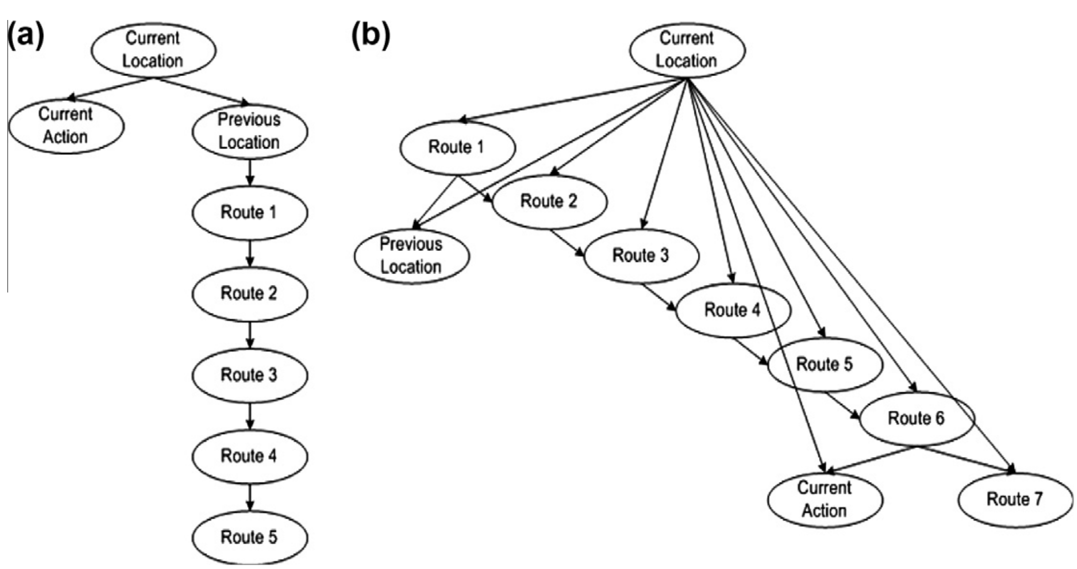


Fig. 1. GBN and TAN location prediction models: (a) learned GBN model using Weka machine learning tool and (b) learned TAN model using Weka machine learning tool.

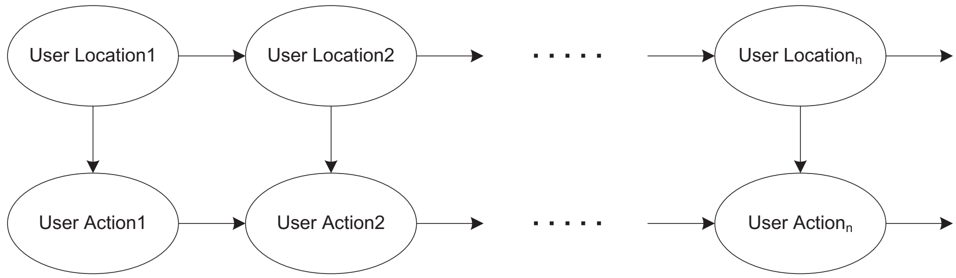


Fig. 2. DBN Location prediction model.

conditional probabilities. In order to avoid zero probabilities, the method of addition of a small number to each cell of sparse conditional probability tables is often employed (Hu, 1999). Often called a flattening constant (denoted by α), this number can be added either only to empty cells, or to all cells in the table. After adding α to cells (either only empty cells or all cells), conditional probabilities are recomputed. While different choices of α have been proposed (e.g., adding 1/2 to all cells or adding $1/D$ to empty cells where D is the total number of cells), we chose to reevaluate conditional probabilities after adding 1 to all cells in the conditional probability tables for the DBN location prediction model.

In the testing phase, the DBN location prediction model begins to make a prediction about the user's location at time 1 by creating slice 1. In slice 1, only two nodes exist: User Action 1 and User Location 1, which comprise all of the available information at the time point. Therefore, we set the current action as the evidence for the node User Action 1 and compute the posterior probability *P*(User Location 1|User Action 1 = observed current action) by using a belief update algorithm, of which any belief update algorithms can be used for propagating beliefs through the DBN. The DBN location prediction model subsequently chooses the location with the maximum posterior probability value as the most probable user location at time 1. At time 2, the DBN location prediction model creates another slice for time 2. At this point in time, the DBN location prediction model consists of two slices for times 1 and 2. The evidence that now exists is the user action and location at time 1 and the user action at time 2. From this information, the DBN prediction model makes a prediction about the user location at time 2 by setting this information as evidence and computing the posterior probability *P*(User Location 2|User Location 1 = the observed location at time 1, User Action 1 = the observed action at time 1, User Action 2 = the observed action at time 2). Now, the location with the maximum posterior probability value becomes the most probable user location at time 2. We describe this inference process below.

For i = 1,..., number of folds:
  1. Partition the dataset into training set i and test set i
  2. Estimate the following prior and conditional probabilities using training set i:

$$P(L_1), P(L_t|L_{t-1}), P(A_t|L_t), P(A_t|A_{t-1}),$$

where A_t and L_t represent user action and location at time t, respectively.
  3. For j = 1,..., number of time slices:
    3.1 Create a slice for time j
    3.2 Set the evidence that has been observed so far:
    If j==1 then:

$$A_1 \leftarrow a_1$$

    Else:

$$A_1 \leftarrow a_1 \quad L_1 \leftarrow l_1$$
$$\cdots \quad \cdots$$
$$A_j \leftarrow a_j \quad L_{j-1} \leftarrow l_{j-1}$$

    3.3 Compute the following posterior probability by using any DBN belief update algorithm:

$$P(L_j|l_{1:j-1}, a_{1:j}),$$

where the action sequence $a_1, a_2, \ldots, a_j$ is denoted by $a_{1:j}$ and the location sequence $l_1, l_2, \ldots, l_{j-1}$ is denoted by $l_{1:j-1}$.
    3.4 Choose the most probable action L^* as follows:

$$L^* \leftarrow \mathrm{argmax} P(L_j|l_{1:j-1}, a_{1:j})$$

  4. Output the predicted location sequence.

---

4. Evaluation

In a formal evaluation, data were gathered from 336 subjects (undergraduate students at a private university in Seoul, Korea). In order to fully engage participation of subjects, two percent of the subject's total class points were given as extra credit points. There were 125 female and 211 male participants of varying ages. The average age of female subjects was 20.7 years old, while the average age of the male subjects was 22.6 years old.

After filling out a demographic survey, participants were asked to record their daily routines (e.g., what they are doing and where they are at when they perform a particular action) on campus over a period of two days. Recordings were started when the subjects arrived at school and ended when they left school for the day. In order to obtain the dataset, we employed a time diary, which in our case was a handwritten log in which the subject writes the start and end times of each action, the location in which the action was performed, and routes that the subject take in order to arrive at the location. Campus location codes, route codes, and action codes were provided to the subjects in order to help them record their daily routines. Initially, there were 84 distinct actions, 25 distinct routes, and more than 30 distinct locations that subjects could choose from, and our subjects recorded a total of 30 distinct locations. We obtained a total of 672 days worth of activity data. Daily activity data was removed if the length of time recorded was too short (e.g., only one action was recorded) or if subjects recorded a route that they could not take at the current route or location. As a result, 266 out of 672 days of activity data were used in the formal evaluation. The examples of actions, locations, and routes that users can take from were described in Table 1.

Table 1
Examples of possible values of domain variables.

Attribute	Examples of possible values
Action	Attending class/attending a seminar, Preparing for exam/doing homework, Doing things that are not listed, Eating lunch, Talking to friends, Studying class materials/self studying, Eating snack/smoking/having tea time, Doing club activities, Surfing the internet, doing a part time job, buying stuffs, meeting with a professor, etc.
Location	600th anniversary building, basketball court, Bicheondang, business building, central library, Dasan hall of economics, east gate, faculty hall, front Gate, Geumjandi square, Hoam hall, international hall, large playground, law building, Myeongnyundang, Oacknyujeong, outside the campus, rear gate, student union building, Suseon hall, Suseon hall annex, Toegye hall of humanities, Yanghyeongwan, Yurimhoegwan, etc.
Route	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W

Table 2
Evaluation results.

Model	DBN	GBN	TAN	NBN
Average accuracy (%)	72.67	45.88	69.29	55.27

4.1. Results

DBN, GBN, TAN, and NBN were induced from data collected using the method described above. The induced models were evaluated using a tenfold cross-validation. For the DBN model, the structure was fixed; conditional probabilities were learned. For the GBN, TAN, and NBN models, the structure and conditional probabilities were learned. Table 2 reports the overall results of DBN, GBN, TAN, and NBN location recognition models. The percentages refer to correctly classified instances. The highest performing model was a DBN location recognition model and the second highest performing model was a TAN location recognition model. The DBN model outperformed all other three models (F-statistics = 29.65, $p < 0.00001$). The accuracy of GBN and NBN seems low compared to the accuracy of DBN and TAN models, but they performed significantly better than chance, which was 3.33% based on the presence of 30 candidate locations. The results suggest that the DBN location prediction model reported on here is able to accurately identify user locations.

The experiment has two important implications for the design of location prediction modeling in ubiquitous computing environments. First, by monitoring a sequence of user locations and actions in the ubiquitous computing environment, induced models can make accurate predictions of forthcoming user locations. Second, using models that can make predictions of user location creates a significant window of opportunity for the ubiquitous environment to take corrective action; context-prediction models offer an improvement over context-aware approaches that predict only current contexts of users.

4.2. Discussion

There are many issues to be discussed from the results above. First, one of the research questions raised in this study was how the ubiquitous decision support system capable of context-prediction mechanism can provide more accurate decision support. In this sense, by using a real data from college students, the proposed DBN was compared with other Bayesian network models such as GBN, TAN, and NBN. Considering the fact that the DBN showed significantly improved accuracy of context-prediction, we conclude that the DBN can be incorporated into the ubiquitous decision support systems as a reliable context-prediction mechanism. With the users' location in the future contexts being predicted accurately, the ubiquitous decision support systems can be used as a reliable source of providing timely and reasonable decision support.

Second, old-aged users and handicapped people could also benefit from the proposed context-prediction mechanism. Where the ubiquitous decision support systems can contribute to social goods most critically is the health information systems for the aged and handicapped people. Since their future locations can be reliably estimated by the proposed context mechanism, those who need care due to their partly handicapped are able to rely on this context-prediction mechanism.

Third, the proposed context-prediction mechanism may also be useful for enhancing the quality of transportation systems. Based on the information about the current location and the goal of a particular user, transportation systems equipped with the proposed context-prediction mechanism may be able to assist drivers more effectively by inferring possible preferred routes and by providing customized route suggestions for drivers, as well as warning the drivers about possible dangers by predicting their future context.

5. Conclusion and Future work

Context prediction is an important problem in ubiquitous computing environments. Accurately predicting user contexts could greatly improve the quality of user satisfaction in every aspect of daily life, particularly in the use of ubiquitous decision support systems. By drawing inferences about user locations, ubiquitous decision support systems should not only automatically detect a user's current situation, but also forecast the user's likely future location. Such location prediction systems will help users make decisions quickly and efficiently by providing the most suitable services.

By utilizing the DBN, this paper has shown that the DBN has potentials to be used as a context-prediction mechanism for the ubiquitous decision support systems. The findings reported here contribute to the growing body of work on context prediction for ubiquitous computing environments. In the future, it will be useful to investigate much richer context information such as a user's intention, or emotional states to predict a user's future context. Secondly, evaluating the resulting models as a runtime component in ubiquitous environments will be an important next step in the development of context-prediction applications. Thirdly, it will be important to investigate location prediction models that can make "early" predictions of user location. Early prediction would allow ubiquitous systems adequate time to prepare for services for a particular future location of the user or to suggest alternative locations before users move to dangerous locations.

Appendix A

A Bayesian network is a direct acyclic graph (DAG) which represents the dependencies among variables and gives a compact representation of full joint probability distributions. Each node in a Bayesian network is annotated with probability distribution. The nodes of the network represent a set of random variables and a direct link connects a pair of nodes. The link from A to B can indicate that A causes B or A has a direct influence on B. The graph structure specifies the conditional independence relationships among variables in the domain. The conditional independence relationships encoded in a Bayesian network can be stated as follows: a node

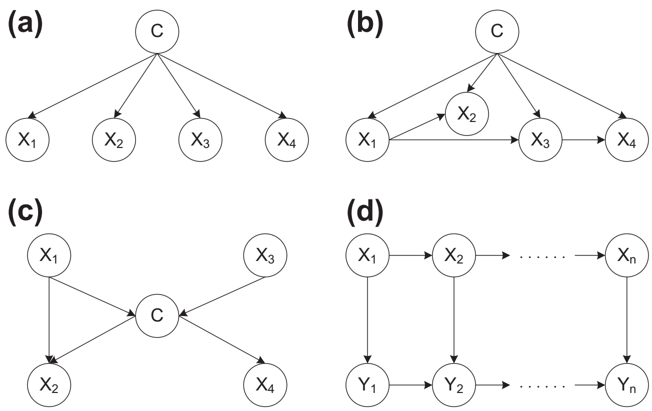


Fig. 3. (a) Naïve Bayesian network. (b) Tree augmented naïve Bayesian network. (c) General Bayesian network. (d) Dynamic Bayesian network.

is conditionally independent of its non-descendant, given its parents; a node is conditionally independent of all other nodes in the network, given its parents, children, and children's parents.

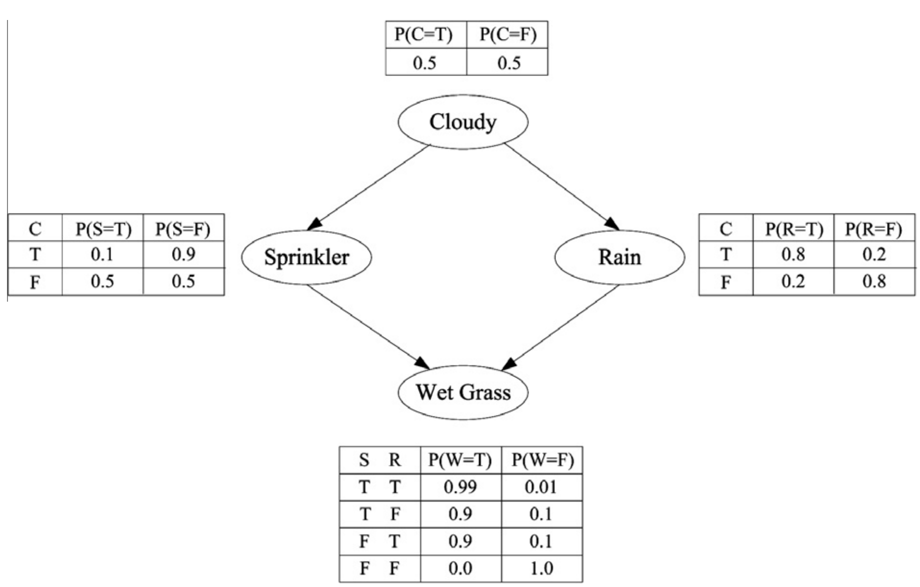
In general, there are four classes of Bayesian networks: Naïve Bayesian networks (NBNs), tree augmented Naïve Bayesian networks (TANs), general Bayesian networks (GBNs), and dynamic Bayesian networks (DBNs) (see Fig. 3). A NBN is the simplest Bayesian network that has one parent node of all other nodes. No other links exist in the network. A TAN is formed by adding directional links between attributes in a NBN. After removing the class node *C*, the attributes should form a tree. A GBN is an unrestricted Bayesian network which treats the class node as an ordinary node. Therefore, the class node can be a child node of some nodes. These Bayesian networks do not provide direct mechanism for representing temporal dependencies among attributes. However most of real-world events are changing over time, which cannot be modeled by static Bayesian networks. A DBN is a Bayesian network that represents attributes with temporal characteristics. A DBN also treats the class node as an ordinary node. The DBN is composed of a sequence of static Bayesian networks. Each static Bayesian network represents the state of variables at different time.

Once the graph structure is constructed, we need to specify a conditional probability table (CPT) which lists the probability that the child node takes given each combination of values of its parents. Consider the following example, in which all nodes are binary (see Fig. 4). The topology of the GBN shows that the event "grass is wet" depends on whether the water sprinkler is on or off and it is raining or not. The event "the water sprinkler is on" depends only on whether it is cloudy or not. From the conditional independence relationships encoded in the network, we can see that the node "Wet Grass" is conditionally independent of "Cloudy", given "Sprinkler" and "Rain". The strength of the relationship among variables is represented as the probability values in the table. For example, we see that $P(\mathrm{W = true|S = true,\ R = false}) = 0.9$ because if the water sprinkler is on or it is raining, then it is likely that grass is wet.

Now, you may have the following question: "Where the graph structure and CPT of each variable come from?." Both the structure of the network and the CPT of each variable can be manually specified by a domain expert, because sometime, it is easy for a domain expert to decide what direct influences exist in the domain and CPTs as well. However, in most domains, the task of defining the topology of the network is too complex for humans and in fact the most challenging task in dealing with Bayesian networks is defining the network structure. In this case, the network structure and parameters of each CPT must be learned from data. We first describe how parameters of each CPT can be learned, given the structure. Then, we introduce learning the structure of Bayesian networks.

Determining prior and conditional probabilities are crucial steps in constructing Bayesian networks. If one can observe all possible configurations of variables and there are no hidden variables in the network, then computing probability tables is merely counting the number of occurrences of each configuration. However, because all possible configurations of variables typically cannot be observed, conditional probabilities of some variables cannot be easily computed from data. For example, if a network has *N* nodes (variables) and even if each node can have discrete binary values, then, at worst, the total number of possible configuration is 2^N. Therefore, it is often the case that sufficient data is not available to learning the conditional probabilities. To avoid zero-probabilities, adding a small number into each cell of sparse conditional probability tables is often employed (Feng et al., 2009).

Learning the structure is much more difficult than learning parameters. If data is missing or some of the nodes are hidden



P(C=T)	P(C=F)
0.5	0.5

C	P(S=T)	P(S=F)
T	0.1	0.9
F	0.5	0.5

C	P(R=T)	P(R=F)
T	0.8	0.2
F	0.2	0.8

S	R	P(W=T)	P(W=F)
T	T	0.99	0.01
T	F	0.9	0.1
F	T	0.9	0.1
F	F	0.0	1.0

Fig. 4. A GBN, showing both the network structure and the conditional probability tables (Russell & Norvig, 2003).

(not observable), then learning is much more difficult. However, structure learning is essential because of its enormous usefulness in various application areas. Structure learning is useful when prior knowledge is unavailable and we want to discover underlying causal or informational relationships to infer knowledge about the domain. Structure learning is the model selection process of choosing among possible models or hypotheses based on an objective function. An objective function measures how well the model can fit the data. Because structure learning is inducing a directed acyclic graph that best explains the given data, the number of possible acyclic graphs given N variables is super-exponential in N (Murphy, 2002). Thus, designers of Bayesian networks often use heuristics to avoid examining all possible structures or they begin with an initial proposed structure. The methods of inducing Bayesian networks from data include conditional independence (CI) based algorithms and search-and-scoring based algorithms (Heckerman, 1995). The CI based approach is based on carrying out several conditional independence tests on the data and building a Bayesian network which agrees with the conditional independence test results. Examples of this approach include Wermuth–Lauritzen algorithm (Wermuth & Lauritzen, 1983), boundary DAG algorithm (Pearl, 1988), SRA algorithm (Srinivas, Russell, & Agogino, 1990), constructor algorithm (Fung & Crawford, 1990), and PC algorithm (Spirtes, Glymour, & Scheines, 1993). Search-and-scoring approaches start with a graph with no edges and then use some search algorithm to add an edge to the graph. They use some scoring criteria to see if the new structure is better than the old one. If it is better, they keep the new one and try to add another edge. Examples of this approach are Chow–Lie tree construction algorithm (Chow & Liu, 1968), Rebane–Pearl polytree construction algorithm (Rebane & Pearl, 1987), and K2 algorithm (Cooper & Herskovits, 1992). Once the graph structure and parameters of each CPT are specified, we can answer all possible inference queries. For example, consider the network shown in Fig. 4. Suppose we observe the fact that grass is wet. We could then ask which one is more likely to be the cause for this: either it is raining or the water sprinkler is on.

A lot of Bayesian software tools provide abilities to learn parameters and/or structure. Some Bayesian tools include API (Application Program Interface) so that users can integrate the Bayesian program into their code. Some tools are free and others are not. Most commercial tools have free versions which are restricted in various ways. One of the widely used Bayesian software is Hugin which was developed at the University of Aalborg. Hugin provides functionality of both parameter and structure learning. The Hugin API is available for the languages C++. Kevin Murphy conducted a survey of Bayesian software tools. The survey covers basic feature of each software such as platform, GUI, API, whether the software allow parameter learning, structure learning or both, etc. The detail of the survey can be found at http://www.cs.ubc.ca/~murphyk/Bayes/bnsoft. html. Google's list of Bayesian tools is also available at http://directory.google.com/Top/Computers/Artificial_Intelligence/Belief_Networks/Software/.

References

Anagnostopoulos, T., Anagnostopoulos, C., Hadjiefthymiades, S., Kyriakakos, M., & Kalousis, A. (2009). Predicting the location of mobile users: A machine learning approach. In *The ACM international conference on pervasive services* (pp. 65–72).

Brdiczka, O., Reignier, P., & Crowley, J. (2007). Detecting individual activities from video in a smart home. In *11th International conference on knowledge-based and intelligent information and engineering systems* (pp. 363–370).

Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory, 14*(3), 462–467.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*(4), 309–347.

Davison, B. D., & Hirsh, H. (1998). Predicting sequences of user actions. In *AAAI/ICML workshop on predicting the future: AI approaches to time-series analysis* (pp. 5–12).

Feng, Y., Teng, T., & Tan, A. (2009). Modeling situation awareness for context-aware decision support. *Expert Systems with Applications, 36*(1), 455–463.

Fung, R., & Crawford, S. (1990). Constructor: A system for the induction of probabilistic models. In *Proceedings of the 7th national conference on artificial intelligence, Boston, MA* (pp. 762–769).

Heckerman, D. (1995). A Bayesian approach to learning causal networks. In *Proceedings of the 11th conference on uncertainty in artificial intelligence, Quebec, Canada* (pp. 285–295).

Hong, J., Suh, E., & Kim, S. (2008). Context-aware systems: A literature review and classification. *Expert Systems with Applications, 36*(4), 8509–8522.

Hu, M. (1999). Model checking for incomplete high dimensional categorical data. Ph.D. Dissertation. Los Angeles, CA: University of California.

Huýnh, T., Fritz, M., & Schiele, B. (2008). Discovery of activity patterns using topic models. In *10th International conference on ubiquitous computing* (pp. 10–19).

Hwang, K., & Cho, S. (2009). Landmark detection from mobile life log using a modular Bayesian network model. *Expert Systems with Applications, 36*(10), 12065–12076.

Kim, E., Helal, S., & Cook, D. (2010). Human activity recognition and pattern discovery. *IEEE Pervasive Computing, 9*(1), 48–53.

Laasonen, K., Raento, M., & Toivonen, H. (2004). Adaptive on-device location recognition. In *2nd International conference on pervasive computing* (pp. 287–304). Springer-Verlag.

Murphy, K. (2002). Dynamic Bayesian networks: Representation, inference and learning. Ph.D. Dissertation. Berkeley, CA: University of California.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press.

Perl, J. (2004). A neural network approach to movement pattern analysis. *Human Movement Science, 23*(5), 605–620.

Petzold, J., Pietzowski, A., Bagci, F., Trumler, W., & Ungerer, T. (2005). *Prediction of indoor movements using Bayesian networks*. *LNCS* (Vol. 3479). Heidelberg: Springer.

Rashidi, P., Cook, D., Holder, L., & Schmitter-Edecombe, M. (in press). Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering*.

Rebane G., & Pearl, J. (1987). The recovery of causal polytrees from statistical data. In *Proceedings of the 3rd conference on uncertainty in artificial intelligence, Seattle, WA* (pp. 222–228).

Singla, G., Cook, D., & Schmitter-Edgecombe, M. (2010). Recognizing independent and joint activities among multiple resident in smart environments. *Ambient Intelligence and Humanized Computing Journal, 1*(1), 57–63.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search*. Berlin: Springer-Verlag.

Srinivas, S., Russell, S., & Agogino, A. (1990). Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information. In *Proceedings of the 5th annual conference on uncertainty in artificial intelligence, Amsterdam, The Netherlands* (pp. 295–308).

Wermuth, N., & Lauritzen, S. (1983). Graphical and recursive models for contingency tables. *Biometrika, 70*(3), 537–552.

Witten, I., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann.