AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2006

Learning and Using Bayesian Networks for Diagnosis and User Profiling

Francisco J. Cantu
Tecnológico de Monterrey

Luis E. Garza
Tecnológico de Monterrey

Armando Robles
Grupo TCA

Ruben Morales
Tecnológico de Monterrey

Learning and Using Bayesian Networks for Diagnosis and User Profiling

Francisco J. Cantu-Ortiz
Tecnológico de Monterrey
fcantu@itesm.mx

Luis E. Garza-Castañón
Tecnológico de Monterrey
legarza@itesm.mx

Armando Robles-Pompa
Grupo TCA
arobles@grupotca.com

Ruben Morales-Menendez
Tecnológico de Monterrey
rmm@itesm.mx

ABSTRACT

We describe various industrial and business applications of a common approach for learning the probability distributions and the structure of Bayesian networks. After describing the Bayesian learning theory, we explain how the parameters learned with this method can be used for prediction tasks in various application domains. In the first domain we learn the structure and the probability distribution of a dynamic Bayesian network to diagnose faults in a generic electrical power distribution network. In the second domain, we learn the probability distributions of a Bayesian network in a multi-agent system for the profiling of users in the procurement and contracting for the USA defense acquisition community. In the third application we describe a help-desk on-line information system for managing a hotel chain using Bayesian reasoning to assign the best technician to answer client requests. With these applications we show how a theoretical Bayesian framework can be integrated into on-line information systems and scaled-up to solve real world problems.

Keywords

Diagnosis, user profiling, data mining, Bayesian learning

INTRODUCTION

Bayesian probabilistic inference has become a common approach for representing and reasoning with uncertain knowledge in multi-agent and intelligent systems (Pearl, 1988). Bayesian inference uses a Bayesian network representation to make inferences about how likely it is that a hypothesis that explains a phenomenon will be true, given certain evidence gathered from the environment. A Bayesian network is a representation technique of uncertain knowledge. The nodes of the network represent random variables and the arcs represent dependencies among variables**.** The dependencies in a Bayesian network may represent causal relationships, where the arrows point from causes to effects. Bayesian inference is commonly applied for diagnostic, classification, prediction, information retrieval and user profiling tasks, among others. In diagnosis, symptoms are observed and we speculate about possible causes using a kind of evidential inference called abduction. In abductive reasoning, we observe certain evidence, and from that evidence, we hypothesize about the possible causes (Poole, 1998). Some hypotheses (causes) are more likely that others, and this is modeled by defining a probability distribution over the hypotheses. Bayesian inference is then used to find the most probable cause (hypothesis) given the evidence. In classification tasks, we are given a set of classes, a set of attributes that characterizes the classes, and for each attribute, a finite domain from which each attribute takes its values. The classes follow a probability distribution which indicates which classes are more likely that others. Each of the attributes is characterized by a probability distribution over the values it takes. When a new case is presented (a case is given by assigning a value to each of the attributes), we want to assign the case a class. We use Bayesian inference to find the probability distribution of this case membership with respect to the classes. We assign the case the class with the highest probability. In a prediction task, we model a problem with a Bayesian network where the arrow points in the causal direction from causes to effects. When we observe some evidence in the environment, we reason abductively (from effects to causes) to find the causes that explain the observations, and once the cause is identified, we reason in the causal direction (from causes to effects) in order to predict the values of other variables that are not observed in the environment, but that are true in that environment. For instance, if we observe fever in a patient, we may abduct that the patient has a blood infection, and if that is the case, we may predict the presence of a virus in the patient's blood. In information retrieval, we want to locate the information that is relevant to answer a query from a user. We fetch the information that is more likely to answer the current question. Another important application of prediction with Bayesian networks is in user profiling. The objective is to design a network that models user habits so that we can predict user's preferences from observed data of user's behavior. Thus, in the diagnosis, classification, prediction tasks, information retrieval or user profiling tasks, we find that the hypothesis (causes) are noisy and that Bayesian inference is able to deal with such environments, by using robust reasoning methods, founded in the well-known, mathematical principles, provided by probability theory.

Bayesian networks and Bayesian inference are used as a knowledge representation as well as an automated reasoning method for uncertain domains, respectively. In one approach to Bayesian modeling, human experts are responsible for designing the network structure by identifying the variables and finding the dependencies among the variables, and also, they are responsible for judging and calculating the conditional probabilities associated with each node. In recent years, attempts have been made to automate both parameters, the design of the network structure, as well as the calculation of the conditional probabilities (Cooper, 1992; Buntine, 1994). This automation process is known as Bayesian learning. Thus, we want to learn these parameters in order to facilitate the designer the use of Bayesian networks in designing intelligent systems for various tasks.

This paper is organized as follows: section 2 describes Bayesian networks; section 3 explains how to learn the two Bayesian parameters: the probability distributions given a network structure, and the structure of a Bayesian network from a database; section 4 explains how a dynamic Bayesian network can be learned for diagnosing faults in an electrical power distribution network; section 5 describes the learning of probability distributions for profiling users in the procurement and contracting for the USA defense acquisition community; section 6 presents a Bayesian on-line help desk information systems for hotel operations; section 7 presents the conclusions

BAYESIAN NETWORKS

The *Bayes theorem* allows us to calculate the posterior probability distribution of a hypothesis given a set of observations (data) from the environment. The Bayes rule is:

$$P(hyp \mid data, K) = P(data \mid hyp, K) \times P(hyp \mid K) / P(data \mid K)$$

K represents background knowledge that has been acquired from previous observations or inferences, and *data* is the actual observation. The likelihood $P(data \mid hyp, K)$ is the probability that the hypothesis would have produced these data. It is high when the hypothesis function is a good fit of the data, and it is low when the hypothesis function would have predicted different data. The prior $P(hyp|K)$ encodes the learning bias. This factor is needed in order to bias the theory towards simple hypothesis functions. The denominator $P(data|K)$ is a normalizing constant to make sure that the probabilities sum to one. If data arrives sequentially, then the Bayes rule gives us a principled way of updating the posterior probability of the hypothesis given the evidence provided in *data*, an given that we have stored in *K* previous knowledge. Thus, $P(hyp \mid d1, d2,..., dn, K1, ..., Km)$, where *di* represents the data elements and *Kj* represents the background knowledge elements, can be calculated using a generalized version of the Bayes rule. The calculation of the right-hand side of the rules becomes computationally intractable. However, a *Bayesian network* facilitates these calculations.

Technically speaking, a Bayesian network is an *acyclic directed graph* (DAG), where nodes denote random variables and arcs represent dependencies between the variables. Associated with each node in the network there is a *Conditional Probability Table* (CPT) that gives the probability that the random variable associated to that node takes a given value, given its parents. Using these CPTs, an efficient Bayesian inference algorithm exists that takes advantage of the independence assumptions modelled by a Bayesian network, to avoid the combinatorial explosion and the intractability that arises in calculating the joint probability distribution, when the number of random variables is large. *The independence assumption embedded in a Bayesian network is that each random variable is independent of its non-descendants given its parents.* Let *x1,...,xn* be a set of random variables in a given domain, and let $P(x1,...,xn)$ be their joint probability distribution. If we totally order the variables and apply the rule-chain, then we have:

$$P(x1,...,xn) = P(x1)\, P(x2 \mid x1)\, P(x3 \mid x1, x2) \ldots P(xn \mid x1, \ldots ,xn\text{-}1)$$

For each random variable *xi*, we assume that there is some minimal set $\Pi xi \subseteq$ *x1,...,xn}* such that $P(x1 \mid x1, \ldots , xi\text{-}1) = P(xi \mid \Pi xi)$. That is, once we know the values of the variables in Πxi, knowing the values of other predecessors of *xi* in the total ordering, will not change our belief in *xi*. The elements of the set Πxi are known as the parents of the variable *xi*. We say that

xi is conditionally independent of its predecessors given its parents. We can create a graph where there is an arc from each parent of a node into that node. Such a graph, together with the CTPs given by $P(xi \mid \Pi xi)$ is known as a *Bayesian network* (BN).

LEARNING BAYESIAN PARAMETERS

In supervised inductive learning we want to approximate an unknown function *f*. We are given a set of examples *(x, f(x))* where *x* is a value and *f(x)* is the value returned by the unknown function *f* when applied to *x*. The supervised inductive learning problem consists of finding a function *h* that approximates the unknown function *f*, given a set of cases of *f* in a database *D*. The function *h* is called a hypothesis. There can be many hypotheses that approximates *f*. The set of these hypotheses is known as the *hypotheses space*. Finding a hypothesis that approximates *f* can be seen as a search problem over the hypotheses space. The database of cases *D* is split into two sets--the *training set* of cases which is used to construct the hypothesis, and the *test set*, which is used to test an hypothesis, once it has been built. Since we know the value *f(x)* for each case *x* in the test set, we can compare *f(x)* against *h(x)* to see how good the hypothesis *h* is. *In Bayesian learning, instead of constructing a hypothesis h as in most learning systems, a posterior probability distribution is constructed over the space of possible hypotheses*. There are at least two different approaches that can be taken in Bayesian inference. One is to choose the hypothesis that is most likely given the data. The other approach is to average over all hypotheses based on their posterior probabilities. Frequently, the structure and the probability distributions in the CPTs are either known or calculated by experts. However, when constructing a Bayesian network we can *learn* these two parameters.

Learning the probability distribution

Let's define a *BN* by a set of random variables $X=\{X1, X2, ..., Xn\}$ and a network structure *S* defining a graph of conditional dependencies among the elements of *X*. Each variable *Xi* takes one of a possible set of *m* values *xi1, ..., xim*. Let *D* be a database of *cases* $D = \{C1, C2, ..., Cl\}$ where each case $Cj = \{x1j, ..., xmj\}$ is a set of entry values. We assume that the cases are mutually independent. Given a network structure *S*, the task is to *learn* the conditional probabilities defining the dependencies in the *BN*, from the database *D*. We regard the conditional probabilities as a set of parameters $\theta = \{\theta 1, ..., \theta k\}$ so that the probability distribution for each case is:

$$P(X=Cj \mid \theta) = \Pi(i=1 \text{ to } n)\, p(xij \mid pa(xi), \theta i)$$

θi parameterizes the probability of xij given the parent configuration $pa(xi)$. We want to assess these parameters θi induced by the database *D*, over the known network structure *S*. The classical statistical parameter estimation provides the basis to learn these parameters. When the database is complete, the common approach is Maximum Likelihood which returns the parameter values that make the database most likely. Given the values of *X* in the database *D*, the joint probability of *D* is:

$$P(D \mid \theta) = \Pi(j=1 \text{ to } l)\, p(X=Cj \mid \theta)$$

This is a function of θ only, usually called the likelihood function $l(\theta)$. The *Maximum Likelihood* (ML) estimate of θ is the value which maximized $l(\theta)$. For discrete variables, the ML estimates of the conditional probabilities are the observed frequencies of the relevant cases in the database. Let $n(xij \mid pa(xi))$ be the observed frequency of cases in the database with xij, given the parent configuration $pa(xi)$, and let $n(pa(xi))$ be the observed frequency of cases with $pa(xi)$. The ML estimate of the conditional probability of $xij \mid pa(xi)$ is simply $n(xij \mid pa(xi)) / n(pa(xi))$.

The *Bayesian approach* to the learning of probability distributions extends the classical statistical parameter estimation technique in two ways: (1) the set of parameters θ are regarded as random variables, and (2) the likelihood is augmented with a prior $\pi(\theta)$ representing the observer's belief about the parameters before observing any data. Given the cases of the database, the prior density is updated in the posterior density using Bayes rule:

$$\pi(\theta \mid D) = \pi(\theta)p(D \mid \theta) / p(D), \text{ where } P(D) = \int \pi(\theta)p(D \mid \theta)\, d\theta$$

The Bayesian estimate of θ is then the expectation of θ under the posterior distribution. Common assumptions in the Bayesian approach to learn probabilities with discrete variables are that the parameters (1) are independent from each other, and (2) have a Dirichlet distribution, which simplifies to a *Beta* distribution for Boolean variables.

If the database is complete, that is, there are no missing data, the posterior distribution of the parameters can be computed exactly using standard conjugate analysis. When some of the entries in the database are missing, we have to face a set of possible complete databases, one for each possible value of the variable for which the datum is missing. Several approaches have been developed for dealing with the situation of incomplete databases (Spiegelhalter and Lauritzen, 1990)

Learning the structure of Bayesian networks

There are at least two approaches to learn the structure of a Bayesian network: the information content approach, where the dependencies are derived based on the amount of information carried between the variables, and the Bayesian approach. The system *Power Constructor* (Cheng, 1999) implements the information content approach. Here we describe the Bayesian approach, which has been implemented in systems *like Bayesian Knowledge Discoverer.*

Suppose we are given a database of n cases $D=\{x1, ..., xn\}$ from which we wish to learn a structure S of conditional dependencies among the variables in the database. If $p(S)$ is our prior belief about a particular structure S, we can use the information in the database D to compute the posterior probability of S given the data:

$$P(M \mid D) = p(M, D) / p(D)$$

We choose the structure with the highest posterior probability. When we compare two candidate structures $S1$ and $S2$, we choose $S1$ if the Bayes factor $P(S1, D) / P(S2, D)$ is greater that 1. $P(S, D)$ can be easily computed if the conditional probabilities defining S are generated as random variables θijk whose prior distribution represents the observer's belief before seeing any data. The joint probability of a case xk can then be written in terms of the random vector $\theta=\{\theta ijk\}$ as:

$$P(xk \mid \theta) = \Pi(i=1 \text{ to } I)\ \theta ijk$$

This parameterization of the probabilities defining S allow us to write:

$$P(S, D) = \int p(S, D, \theta)\, d\theta = p(S) \int p(\theta \mid S)\, p(D \mid \theta)\, d\theta$$

Where $p(\theta \mid D)$ is the prior density of θ, and $p(D \mid \theta)$ is the sampling model. The integral is defined in the parameter space whose dimension depends on the complexity of the network. A solution in closed form exists if, (1) the database is complete, (2) the cases are independent given θ, (3) the prior distribution of the parameters is conjugate to the sampling model $p(D \mid \theta)$, and (4) the parameters are independent. Details of the calculation can be found in (Cooper and Herskovitz, 1992). (Ramoni and Sebastiani, 2000) have developed a method to learn Bayesian structures when the database is incomplete.

Bayesian classifiers

Classification is a common task performed by knowledge discovery systems. It consists of assigning a class label to a set of unclassified cases. Classification typically involves two steps: first, the system is trained on a set of data and then it is used to classify a new set of unclassified cases. A Bayesian classifier is trained by estimating the conditional probability distribution of each attribute, given the class label, from a data base. The estimation of these probabilities is analogous to the calculation of the weights is a neural network. A case is classified from its set of attribute values, by computing the posterior probability of each class label given the attribute values, using the Bayes rule. The case is then given the class with the highest posterior probability. The simplifying assumption underlying Bayesian classifiers is that the classes are mutually exclusive and exhaustive and that the attributes are conditionally independent once the class is known. These assumptions have been regarded too strong and have dismissed Bayesian classifiers as *naive*, useful just for evaluating more sophisticated techniques. But despite these theoretical limitations, recent empirical evaluations have found Bayesian classifiers surprisingly accurate (Friedman et al., 1997). A Bayesian classifier is defined by a set C of classes and by a set A of attributes. We denote a generic class by cj, and a generic attribute by Ai. The Bayesian assumptions about mutually exclusive and exhaustive classes and conditionally independent attributes are made. This way, the set of classes C can be treated as a stochastic variable taking one of the values ci with a probability distribution that represents the unknown state of the world. The database of cases are used to determine the probabilities $P(cj)$ and $P(Ai \mid cj)$ for each attribute Ai. These probabilities are determined by counting the number of instances. All of the attributes depend on their class only, and no connections among attributes are allowed. Figure 1 shows the Bayesian network assumed by a *naive* Bayesian classifier.

Suppose we observe a new case with $A1=v1, A2=v2, ..., Ak=vk$. We use the Bayes rule to determine the posterior probability of the class cj of the new case conditioned on the values of the attributes, as follows:

$$P(cj \mid A1=v1,..., Ak=vk) = P(A1=v1,...,Ak=vk \mid cj)P(cj) / P(A1=v1,..., Ak=vk)$$

Using the independence assumption this simplifies to:

$$P(cj \mid A1=v1,..., Ak=vk) = P(A1=v1 \mid cj) \text{ x...x } P(Ak=vk \mid cj)P(cj) / P(A1=v1,..., Ak=vk)$$

The values $P(A1=vi \mid cj)$ are obtained from the CPTs. The denominator P(A1=v1,..., Ak=vk) is a normalizing factor to ensure that the probabilities add to one. Ramoni and Sebastiani proposed a *robust* Bayesian classifier called *ROC* that is able to deal with incomplete databases, with no assumption about the pattern of missing data (Ramoni and Sebastiani, 1999).

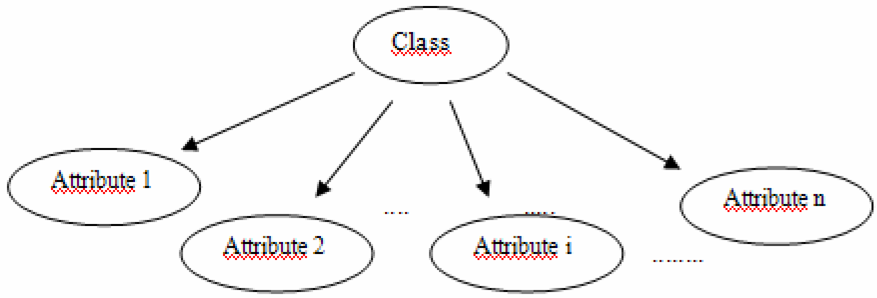

Figure 1 Bayesian network for a naive Bayesian classifier

BAYESIAN DIAGNOSIS IN POWER DISTRIBUTION NETWORKS

Bayesian networks have been frequently applied for the task of fault diagnosis. Diagnosis is seen as an evidential reasoning task, where some parts of the system are observed and we want to make inferences about other hidden parts of the system. We use Bayesian networks to model a problem domain with either a causal model, or an evidential model, or both. We use this model for doing either evidential or causal reasoning for diagnosing system faults as shown in figure 2.

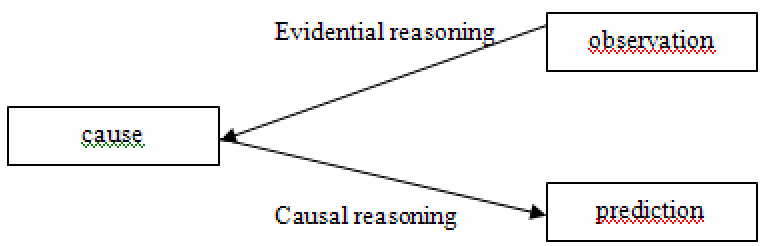

Figure 2.Causal and evidential reasoning in diagnosis

Evidential and causal reasoning are combined for diagnosis. For instance, a medical doctor may observe symptoms in a patient, determine possible diseases, and predict symptoms which have not manifested. The doctor then looks for the presence of these new symptoms following a perception cycle. If the Bayesian network is used as a causal model then we use abduction to reason from observations to causes and we use model-based reasoning to make predictions in the causal direction. Poole proposed the *Independent Choice Logic* (ICL), a framework for modeling multiple agents under uncertainty. ICL combines Bayesian networks and first-order logic representations, as well as influence diagrams, dynamic systems, decision making and game playing (Poole, 1997). One of the applications of ICL is in multiple fault diagnosis. A methodology based on a dynamic version of ICL was developed by Garza (Garza, 2001) and applied it in diagnosing faults in electric power transmission networks (Garza et al., 2000). The methodology comprises three steps. In the *first step*, the problem is modeled causally, using the ICL representation, a set of Horn-logic rules encoding probabilistic information. In the *second step*, the ICL system is used to generate a set of explanations of possible causes from observed symptoms. Since the search space of explanations grows exponentially in the number of variables, heuristics based on probability maximization, entropy and knowledge of the domain are used to prune the search space. In the *third step*, a causal reasoning is conducted by using *Dynamic Bayesian networks* to analyze continuous signals, to collapse the remaining explanations from step 2, into a single explanation that explains the observed symptoms. This methodology has been applied to the diagnosis of faults in power transmission network, of a complexity equivalent to an electric network in a city area. This problem is documented by IEEE as a benchmark problem in system diagnosis. Other problems in diagnosing industrial machinery are also tackled with this methodology. (Garza et al., 2005).

BAYESIAN USER PROFILING IN THE MACS SYSTEM

Other important application area of Bayesian networks is user profiling (user modeling). The objective is to develop a model of user habits and use this model to predict user behavior, from observed user action. This is also an evidential and causal reasoning task. Thus, the Bayesian approach to user profiling is similar to the Bayesian approaches for classification and diagnosis. Heckerman and Horvitz describe the use of a Bayesian approach to modeling the relationship between words in a query from a user who asks for assistance, and the informational goals of the user (Heckerman and Horvitz, 1998). Their work was used by the Microsoft Office to help Excel users when using a spreadsheet. Rather than using key-words for which a full list of synonymous is likely not to be available, they proposed a probabilistic approach for diagnosing user's problems given a query, by inferring a probability distribution over concepts in the query.

Following these ideas, we describe Bayesian learning for user profiling in the system *Multi-Agent COTR System for Defense Contracting* (MACS) (Liebowitz et al, 2000). The *MACS* system is a multiagent-based system for assisting Contracting Officers´ Technical Representatives (*COTRs*) in the pre-award phase of defense contracting and procurement. The system architecture consists of a user agent and a set of five specialty agents. The user agent interacts with the *COTR* and broadcasts a *COTR*'s query to the various specialty agents, who analyze the query so that hopefully, one of them will provide an answer. We enhance the functionality of the user agent by having it to "know" more about the *COTR* and by having it collaborate with the specialty agents in deciding how to answer a query, instead of just broadcasting the query. The former, (knowing more about the *COTR*) is a user profiling function that can be obtained by Bayesian learning, and is described in this section. The latter, (cooperating with the specialty agents) is accomplished with a type of Bayesian learning technique for multi-agent systems and is explained in (Cantu, 2002).

All the knowledge and reasoning required to answer the query resides within the specialty agent that means that a specialty domain is independent of the rest of the specialty domains. The actions of the specialty agents comprise mainly the reasoning that is carried out and the knowledge that is inferred in order to answer a query. The main agent's actions are summarized as follows:

(1) *Forms agents*. How to identify the forms needed to complete the contract request based on characteristics of the contract.

(2) *Justification agent*. How to determine whether a justification and approval is required to complete the procurement request

(3) *Evaluation agent*. Determining an evaluation function and its weight parameters or an evaluation criteria

(4) *Synopsis agent*. Under what circumstances is a synopsis required for completion of a procurement request package

(5) *Type of contract agent*. How to classify a contract based on its main attribute values

The learning capability that we add to MACS consists of having the user agent identify the specialty agent probabilistically, by considering the specialty domains as random variables with a probability distribution, given the evidence appearing in the query. This is done by a *naive Bayesian classifier*. We infer user's goals from user's queries in free-text format using a naive Bayesian classifier. The user agent uses a Bayesian model to identify the specialty agent given the evidence appearing in the query. This model shows a causal relationship between the specialty agent domains and the evidence in the query. There is an ontology for the specialty agents that includes a goal. Each specialty agent is described by a set of attributes that are terms that typically appear in a query directed to that specialty agent. The corpus of queries stored in the US Defense Acquisition DeskbookWeb page, provided the information to define the classes, the attributes and finding the necessary probabilities required by Bayesian inference. The structure of the network and the probabilities are learned from this database of queries using the algorithms described in section 2 (Cantu, 2002). The problem of learning and inference in multi-agent systems in similar problems is now tackled with the concept of electronic institutions (Robles et al, 2005b).

BAYESIAN REASONING FOR AN ON-LINE HELPDESK

A Bayesian Reasoning Framework (BRF) is a working environment developed by Robles in which an on-line information system learns relations that exist in data and uses that knowledge to support business decision processes using Bayesian Inference (Robles, 2003). The on-line information system described in this study manages the operation of a chain of hotels in Latin America. BRF was used to replace a manual help-desk sub-system of the hotel management system by an automated on-line help desk. The BRF help desk is responsible for assigning the best technician for each client service request about system operation. The assignment is made by an autonomous Bayesian agent that was modeled using the allocation inference template of Common KADS (Schreiber et al., 2000). BRF uses a multi-layer architecture and includes a Bayesian Reasoning agent for which we implemented the techniques of section 2 for learning the Bayesian probabilities for inferring the best technician allocation. The help desk system uses a Bayesian network whose probabilities are learned by the autonomous agent using Bayesian learning. BRF comprises a three-layer environment with business information systems at the top, a middle-ware Bayesian reasoning techniques server, and a Bayesian reasoning engine at the bottom. The main goal of the helpdesk system is to provide technical support service through the internet to users of the hotel management information system. Each hotel has from 10 to 40 users whose technical support requirements are attended by the helpdesk system. We use 9,000 data records produced by 3 years of normal help desk system operation, 7,000 were used to learn the dependencies involved in the advisors assignation process and 2,000 were used for testing the generated assignations by the inference algorithm. We found that the performance of the help desk based on BRF is at least as good as the performance of its manual counterpart. (Robles et al., 2005a). Extensions of BRF have been used by an on-line hospital management information systems that uses multi-agent systems for modeling electronic institutions (Robles et al, 2005b).

Other applications of Bayesian learning using the theory and algorithms described in this paper have been developed but are not included because of space limitations. Among these applications are chemical process control (Morales et al., 2004a), autonomous robot navigation (Morales et al., 2004b) and user authentication by learning key-stroke patterns (Gutierrez et al, 2002).

CONCLUSION

We have described industrial and business applications of a common approach for learning the probability distribution and the structure of a Bayesian network. We have described how learning the probability distributions and the structure of a Bayesian network can be done from data, and have presented three applications: diagnosing faults in a electrical power distribution network, the profiling of users in the procurement and contracting for the USA defense acquisition community and in an on-line help desk information system for managing the operation of a hotel chain. With these applications we show how a theoretical Bayesian framework can be integrated into on-line information systems and scaled-up to solve real world problems

ACKNOWLEDGMENTS (OPTIONAL)

We acknowledge the support from Tecnológico de Monterrey for the financial aid and facilities in developing this research.

REFERENCES

1. Buntine, W.L. (1994). Operations for learning with graphical models. Journal of Artificial Intelligence Research, Vol. 2, pp. 159-225.
2. Cantu, F. (2002). Reinforcement and Bayesian learning in multi-agent systems: The MACS project. Technical report CIA-RI-042, Center for Artificial Intelligence, ITESM, Mexico
3. Cheng, J. and Greiner, R., (1999). Comparing Bayesian Network Classifiers. In proceedings of the fifteenth conference on uncertainty in artificial intelligence UAI99
4. Cooper, G.F. and Herskovitz, E. (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning, Vol. 9, pp. 309-349.
5. Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. Machine learning.
6. Garza, L.E. (2001). Hybrid systems fault diagnosis with a probabilistic logic reasoning framework, PhD Thesis. ITESM, Mexico
7. Garza, L.E., Cantu.Ortiz, F., Acevedo, S. (2000). A Methodology for Multiple-fault Diagnosis based on the Independent Choice Logic. In Proceedings of IBERAMIA and the Brazilian International Conference on Artificial Intelligence, Springer-Verlag Lecture Notes on Artificial Intelligence, November 18-22, 2000, Sao Pablo, Brazil
8. Garza, L.E., Morales, R. Cantu, F.J. (2005). *A Framework to Design Fault-Tolerant Mechatronics Systems under Uncertainty*. Proceedings of IASTED International Conference on Artificial Intelligence and Applications 2005, Austria.
9. Gutierrez, F. J., Lerma, M., Salgado, L. R., Cantu, F. J. (2002). Biometrics and Data Mining: Comparison of Data Mining-Based Keystroke Dynamics Methods for Identity Verification, MICAI-2002, Lecture Notes in Artificial Intelligence 2313, Springer-Verlag
10. Heckerman, D. and Horvitz, E. (1998). Inferring informational goals from text-free queries: A Bayesian approach. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp 230-237.
11. Liebowitz, J., Ayda, M. Rubenstein-Montano, B., Yoon, Y., Buchwater, J., Imhoff, M., Baek, S., Suen, C. (2000). MACS: A multi-agent COTR System for Defense Contracting. Technical report, UMBC, Baltimore.

12. Morales-Menéndez, R., Francisco J. Cantú, Antonio R. Favela (2004). Dynamic Modelling of Processes using Jump Markov Linear Gaussian Models. Proceedings of the 23º IASTED Conference on Modelling, Identification and Control, Grindelwald, Switzerland, February.
13. Morales-Menendez, R., J.Mutch, F. Guedea, L.E. Garza, F.J.Cantu (2004). Diagnosis in mobile robots using particle filtering algorithms. Proceedings of the IASTED International Conference on Intelligent Systems and Control, Honolulu, Hawaii, USA.
14. Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, USA
15. Poole, D. (1998). "Learning, Bayesian Probability, Graphical Models, and Abduction", Peter Flach and Antonis Kakas, editors, Abduction and Induction: essays on their relation and integration, Kluwer.
16. Poole, D. (1997). The Independent Choice Logic for modelling multiple agents under uncertainty. Artificial Intelligence Vol. 94, pp. 7-56.
17. Ramoni, M., Sebastiani, P., and Crea, A. (2000). Profiling your customers using Bayesian networks. ACM SIGKDD Explorations, 1(2).
18. Robles, A. (2003). Un ambiente de trabajo para sistemas de información basado en razonamiento Bayesiano. Master thesis. ITESM, Mexico
19. Robles, A., Cantú, F.J., Morales R. (2005a). A Bayesian Reasoning Framework for On-line Business Information Systems. Proceedings of the Eleventh Americas Conference on Information Systems, Omaha Nebraska, USA, August 11-14.
20. Robles, A., Noriega, Cantú, F.J., Morales R. (2005b). Enabling Intelligent Organizations: An Electronic Institutions Approach for Controlling and Executing Problem Solving Methods. Lecture Notes on Artificial Intelligence, Springer Vol. 3789, pp 275-286.