A Bayesian Network-Based Expert Query System for a Distributed Database System

Mei-Ling Shyu
University of Miami
Department of Electrical and
Computer Engineering
Coral Gables, FL 33124
shyu@miami.edu

Shu-Ching Chen
Florida International University
School of Computer Science
Miami, FL 33199

chens@cs.fiu.edu

Abstract

A distributed database is a collection of data sources distributed across many computers. It is one logical and centrally managed database stored in multiple physical locations. In order to access data locally and manage data globally, a good *distributed database system (DDBS)* that can supply reliable and up-to-date information is critical. A good *DDBS* should have the functionality such as providing users timely and flexible access to information, providing the personnel tools to analyze the data in a meaningful manner, and allowing the personnel to control the safety and integrity of the data. However, query processing in such a distributed database system is complicated. For example, how to keep data up-to-date when it is physically dispersed and how to propagate information to the whole *DDBS* when an update occurs are some important issues. In response to these issues, we propose an expert query system that uses Bayesian network as its framework to propagate information and to keep data up-to-date for a distributed database system. The goal of this query system is to assist the personnel in maintaining the data dynamically and to assist the users in accessing reliable and timely information.

1 Introduction

Recent technological advances have provided users with the possibility of accessing information through *distributed database systems (DDBSs)*. A distributed database consists of multiple data sources spread across many computers. Due to the complexity of real-world applications, the number of data sources and the volumes of data in the data sources have increased tremendously. By allowing data to be accessed locally and managed globally, a good *DDBS* is able to supply reliable information anytime and anywhere.

In a general-purpose *DDBS*, the tasks are to provide timely and flexible access to information, to provide tools for analyzing the data in a meaningful manner, and to allow the control of the safety and integrity of the data. However, query processing in such a distributed database system is complicated. Beyond the core technology, a number of challenges need to be considered for query processing. These challenges encompass issues such as the consistency and availability of data, how to keep data up-to-date, and how to propagate information in the system. It is highly desirable for users and/or the personnel to be able to access and/or update the individual databases simultaneously and without the burden of having to deal with or be aware of the issues mentioned above.

Toward this end, an expert query system that uses Bayesian networks as its framework to propagate information and to keep data up-to-date for a distributed database system is proposed in this paper. The proposed Bayesian network-based expert query system is also implemented by using the *CLIPS (C Language Integrated Production System)* object-oriented programming language for the services on the lands of an amusement park as an example *DDBS*. The purpose of this expert query system is to assist the personnel and the users for using the *DDBS*. The personnel use this query system to update and maintain the *DDBS*; while the users issue queries to retrieve information of their interests from the *DDBS*. A local area network (LAN) is used to connect the sites of the *DDBS* for the amusement park. The Bayesian network used here is a tree-structured topology that is suitable for the geography of the lands in the amusement park. The up-to-date information is maintained by the personnel via the updating and propagation of the beliefs. Since the beliefs of the nodes in the Bayesian network indicate the availability of the corresponding lands and suggest the visit sequences of other lands to the users, the users can access the most recent and useful information to obtain better schedules of their tours in the amusement park. This shows the feasibility of the proposed query system when it is applied to a *DDBS*.

The paper is organized as follows. The Bayesian belief network is briefly introduced in Section 2. The proposed Bayesian network-based expert query system is presented in Section 3. The conclusions are drawn in Section 4.

2 Bayesian Belief Networks

A Bayesian network (or Bayesian belief network) is a state-of-the-art knowledge representation scheme in real-world intelligent systems dealing with uncertainty [6]. Bayesian networks have been used in many applications such as forecasting, visual recognition, medical diagnosis, forecasting, ship identification from radar image, and device trouble-shooting [1, 2, 3, 4, 5, 7].

Bayesian networks use the richer language of *directed acyclic graphs (DAGs)*, where each node represents a random variable or uncertain quantity, the arcs signify the existence of direct causal influences between the linked variables, and the strengths of these influences are quantified by conditional probabilities [8]. Each node independently receives messages from all of its neighbors (direct causes and direct effects) about the change of beliefs at those nodes, updates its own belief based on these messages, and then sends messages about its new belief to all its neighbors. The process of updating a node's belief is called *fusion* because all messages received are fused together; while the process of sending messages is called *propagation*. This fusion and propagation process continues until the whole network (or the part of it which can be reached from the input nodes) has been updated and the network reaches an equilibrium where each node has its belief revised to reflect the presence of new data. This process can be shown to take only time proportional to the diameter of the network [9, 10]. The fusion and propagation process is adopted to capture the information flows among the sites of a *DDBS* and furthermore, to maintain load balancing in the *DDBS*.

Bayesian methods also provide a formalism for reasoning about partial beliefs under conditions of uncertainty. This proposed expert query system is designed by viewing the belief network not only as a passive code for storing factual knowledge but also a computational architecture for reasoning about the knowledge. This means that the links in the network should be treated as the only mechanisms that direct and propel the flow of data through the process of querying and updating beliefs. While the impact of each new piece of evidence is viewed as a perturbation that propagates through the network via message-passing between neighboring variables, each proposition is assigned a certainty measure consistent with the axioms of probability theory.

In addition, a full specified Bayesian network constitutes a complete probabilistic model of the variables in a domain. It contains the information needed to answer all probabilistic queries about those variables. The queries may be the requests to interpret specific input data or the requests to recommend the best course of action. Interpretation requires instantiating a set of variables corresponding to the input data, and selecting the most likely combination of these hypotheses. The up-to-date information is maintained via the updating and propagation of the beliefs. In our design, the beliefs of the nodes in the Bayesian network are used to indicate the availability of the corresponding sites of the *DDBS*. Therefore, the updated beliefs can be used to suggest the visit sequences of the sites of the system to reflect the most recent and useful information to the users of the system. In the amusement park example, the information can be provided to allow the customers to obtain better schedules of their tours in the amusement park.

3 The Bayesian Network-Based Expert Query System

In this paper, a Bayesian network-based query system is designed to update and propagate information dynamically for a *DDBS*. The proposed expert query system design is based on the traditional diagnostic system that is implemented using *CLIPS* for the services on the lands of an amusement park as an example *DDBS*. The object-oriented paradigm is adopted since the object-oriented programming language fits the needs of the system and can be well-maintained. A local area network (LAN) is used to connect the sites of the amusement park (or the *DDBS*) in the implementation. The potential advantages of using LAN topology are: (1) it provides better response time since only short delays are involved because of the short distances of the sites in the network, (2) it is more flexible

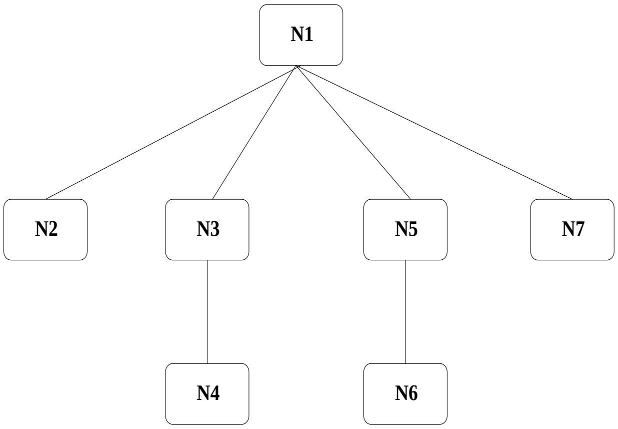

Figure 1: Tree topology for the Bayesian belief network. N1 to N7 represent the nodes (lands in the amusement park) for the network.

to increase new sites in the network if necessary, (3) changes will not cause big conflicts for the whole system. In this LAN, there is a central control site (land 1) that connects to the other six sites in the way that each site connects to its neighbors to form a closed loop from sites 2 to 7 (lands 2 to 7). Each site has a local computer center which controls all the data information in that site.

3.1 Topology

A general tree-structured Bayesian network topology that each node has only one parent and might have several children for this example (as shown in Figure 1). The nodes of the Bayesian belief network are the seven lands in the amusement park and are assumed to be boolean variables, i.e. each node has only two instantiations (**0** or **1**). As can be seen from this figure, the connectivity of the tree structure represents the geography of the park. For example, node N1 that represents land 1 (Main_Land) is the main entrance of the park so the root node of the tree is node N1. Also, any customer who wants to visit land 4 (node N4) needs to go through lands 1 and 4.

In the implementation, each service is represented by a class name and its associated attributes which contain the basic information of that service. The root node of the tree topology is represented by the class *Main* and contains the following attributes:

- LNO - the land number.
- LNAME - the land name.
- π - the π-message of the node.

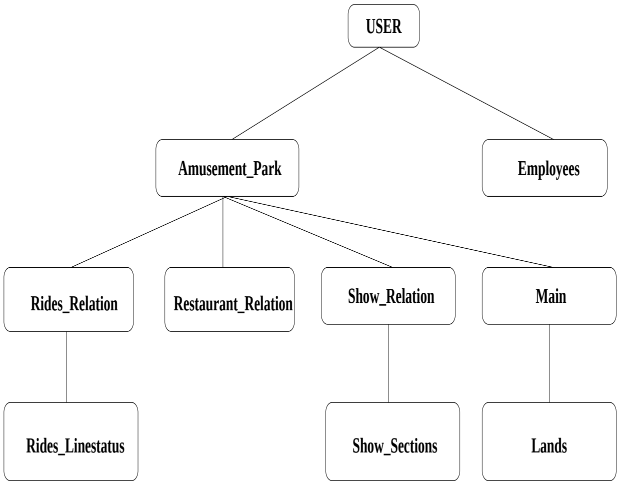

Figure 2: Class hierarchy for the amusement park example.

- λ - the λ-message of the node.
- BEL - the belief of the node. We use the first instantiation of the value BEL to be the belief for the customers to visit the corresponding land of the node.

On the other hand, the interior nodes and the leaf nodes of the tree are represented by the class *Lands* that contain the attributes:

- LNO - the land number.
- LNAME - the land name.
- π - the π-message of the node.
- λ - the λ-message of the node.
- BEL - the belief of the node.
- $\pi2$ - outgoing π-message sent to its children nodes.
- $\lambda2$ - outgoing λ-message sent to its parent node.
- matrix1 - first row of the link matrix of the node.
- matrix2 - second row of the link matrix of the node.

3.2 Class Hierarchy

The class hierarchy for the implementation is shown in Figure 2. Partial instances for each class are also defined to show how this system works. For example, the basic information of the SHOP services in the amusement park is defined as follows.

```
(defclass Shop_Relation (is-a Amusement_Park)
 (slot SPNO        (multiple))
 (slot name        (multiple))
 (slot type        (multiple))
 (slot credit_card(multiple)(default yes))
 (slot open_close (multiple)(default open))
 (slot time        (multiple))
)

(definstances shops
 (SP1 of Shop_Relation
  (LNO 1) (LNAME Main_Land)
  (SPNO 1) (name Main_St_Confectionery)
           (type snacks))
 (SP2 of Shop_Relation
  (LNO 1) (LNAME Main_Land)
  (SPNO 2) (name The_Shadow_Box)
           (type old_fashion))
 . . . .
)
```

3.3 Query Interface

As soon as the system is started, the following main menu is prompted to the users of the system. As can be seen from this main menu, option **1** is for the customers to query information in the system, option **2** is for the maintenance personnel to maintain the system, and finally, option **0** allows the users to exit the query system.

```
WELCOME TO THE EXPERT QUERY SYSTEM!!!
Start the database query system .....
############################
### Select your choice : ###
############################
0 => Exit
1 => Customer query
2 => Database maintenance
```

If option **1** is selected, then the system is mainly used by the customers to retrieve information with the following prompt:

```
********************************
*** Select your query item : ***
********************************
0 => Exit
1 => Shop Information
2 => Ride Information
3 => Ride Linestatus
4 => Restaurant Information
5 => Show Information
6 => Show Reservation
7 => Visit Sequence Information
```

Each option provides different information for different service. For example, option **1** gives the information of the shops in the amusement park such as the shop number, shop name, the land number this shop is located, shop type, whether the shop accepts credit card or not, whether the shop opens or not today, and the shop business hours. Similarly, options **2** to **6** provide information for the rides, restaurants, and shows. Option **7** is the most important service provided by the proposed expert query system. It analyzes the whole system by examining the beliefs of the network and gives the suggested sequences for the customers to visit the remaining lands. Through this query, the customers know the crowded and available situations of the other lands and then can schedule their visit tours effectively.

On the other hand, if the user is a maintenance personnel of the amusement park, then option **2** in the main menu is chosen and the following maintenance menu is popped up :

```
**************************************
*** Select your maintenance item : ***
**************************************
0 => Exit
1 => Shop Information
2 => Ride Information
3 => Ride Linestatus
4 => Restaurant Information
5 => Show Information
6 => Show Section
7 => Employee Information
8 => Bayesian Network Beliefs
```

Each selected item authorizes the personnel the ability to update different databases stored in the system. The main goal is to update the database system as soon as possible such that when the customers do their queries they can get the newest information they need. The first choice prompts for the shop number to be updated. Once the personnel chooses the right shop number, he or she is asked to select the item to be updated such as changing the shop type, the credit card acceptance situation, the open_close information, and the business time of the shop. Again, options **2** to **7** allow the updating of the information related to the rides, restaurants, shows, and employees. The last option is to update and propagate the beliefs of the nodes in the Bayesian network. No matter how many nodes are updated, which node is updated, or which data is changed, the new beliefs of all the nodes are recalculated. Once the beliefs of the network are updated, what the customers get for their queries are the best scheduling precedences for their tours.

3.4 Link Matrix

Each direct link $X \rightarrow Y$ is quantified by a fixed conditional probability matrix M (also called a link matrix), in which the (x, y) entry is given by

$$
\begin{aligned}
& M_{y|x} \approx P(Y = y \mid X = x) \\
= & \left[\begin{array}{cccc}
P(y_1 \mid x_1) & P(y_2 \mid x_1) & \ldots & P(y_n \mid x_1) \\
P(y_1 \mid x_2) & P(y_2 \mid x_2) & \ldots & P(y_n \mid x_2) \\
\ldots & & & \\
P(y_1 \mid x_m) & P(y_2 \mid x_m) & \ldots & P(y_n \mid x_m)
\end{array} \right]
\end{aligned}
$$

Since the Bayesian network is implemented only for propositional logic and each node has only one parent node, the size of the link matrix is 2 × 2 in this example expert query system. To generate the link matrix, two matrices (matrix1 and matrix2) are used to store the first and the second rows of the link matrix. The first value for each row should be provided by the maintenance personnel. Since the summation of each row must be one, the second value of each row can be obtained by subtracting the given value from the value one directly.

For example, if P(N2=0 | N1=0)=p1 and P(N2=0 | N1=1)=p2 are provided by the personnel, then the corresponding link matrix contains the following probabilities, and the first row is stored in matrix1 and the second row is store in matrix2, respectively.

$$M_{n2|n1} = \left[\begin{array}{cc} p1 & 1 - p1 \\ p2 & 1 - p2 \end{array} \right]$$

3.5 Belief Propagation

The belief propagation scheme in trees works in the following manner. Each node must combine the impacts of all the λ-messages obtained from its children nodes, and distribute a separate π-message to each of its children nodes. Note that all the π, λ, $\pi2$, and $\lambda2$ values are normalized each time when there are updated values in the implementation so that the summation of the values is one.

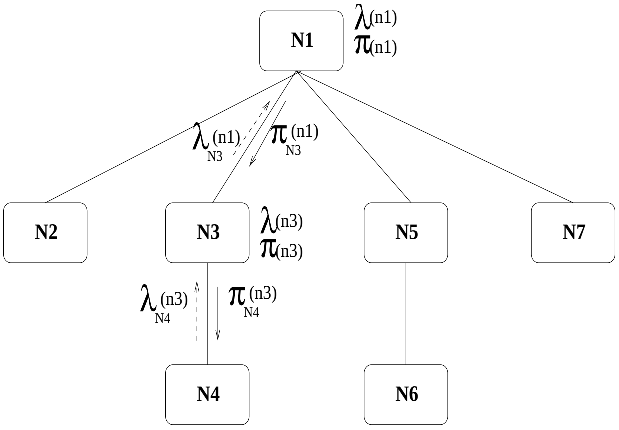

Figure 3: Belief propagation for node N3.

Assume a node X has one parent node (U) and j children nodes $(Y_1, \ldots, Y_j)$, its belief $(BEL(x))$, λ-message $(\lambda(x))$, $\lambda2$-message $(\lambda_x(u))$, π-message $(\pi(x))$, and $\pi2$-message $(\pi_{Y_i}(x))$ are defined as follows.

$$
\begin{aligned}
BEL(X) &= \alpha\lambda(x)\pi(x), where & (1) \\
\lambda(x) &= \prod_{i=1}^{j} \lambda_{Y_i}(x), & (2) \\
\pi(x) &= M_{x|u}\pi_X(u), & (3) \\
\lambda_X(u) &= M_{x|u}\lambda(x), & (4) \\
\pi_{Y_i}(x) &= \alpha\pi(x) \prod_{k \neq i} \lambda_{Y_k}(x). & (5)
\end{aligned}
$$

Here, α is a normalizing constant.

The π-message of the root node is set to be the prior probability of the root variable. The leaf nodes need to be instantiated before the belief propagation process. If a leaf node is instantiated and the jth value of this node is observed to be true, then its λ-message=**(0,...,1,0,...,0)** with **1** at the jth position.

Figure 3 shows part of the belief propagation process for node N3 in Figure 1. In this example, N3 has one parent node N1 and one children node N4. The steps involved in the belief propagation process for node N3 are summarized by specifying the following activities:

1. Belief updating: When node N3 is activated to update its parameters, it simultaneously inspects the $\pi_{N3}(n1)$-message communicated by its parent node and the $\lambda_{N4}(n3)$-message communicated by its children node. The $\lambda(n3)$-message and $\pi(n3)$-message are updated by using Equations 2 and 3. Then, it updates its own belief via Equation 1.

2. Bottom-up propagation: Generate the $\lambda_{N3}(n1)$-message by using Equation 4 and sent it to its parent node N1.

3. Top-down propagation: Generate the $\pi_{N4}(n3)$-message by using Equation 5 and sent it to its children node N4.

The beliefs of the nodes in the Bayesian network are used to indicate the availability of the corresponding lands and give the visit sequences of the other lands to the customers. The first instantiation (i.e. instantiation 0) is assumed to be the belief index for the availability. Of course, when new instantiations of the leaf nodes are given, a new π value of the root node is given, or the values of the link matrix are updated, then the beliefs are updated accordingly. Therefore, the customers can get the newest information and schedule their visit sequences to avoid the crowded situations.

4 Conclusions

In this paper, an expert query system through the updating and propagating of the beliefs of the Bayesian network for a *DDBS* is presented. This proposed expert query system is designed by viewing a belief network not only as a passive code for storing factual knowledge but also a computational architecture for reasoning about that knowledge. The lands of an amusement park are used as an example to demonstrate how the expert query system can be applied to a *DDBS*.

The purpose of this expert system is to assist the personnel and the customers for using the services in the amusement park. The main jobs of the personnel are to maintain the databases and to provide the newest information for the services. While in the customers' point of view, they want to get the most useful information and the best schedules of their tours in the park. Moreover, an overall examining of this expert query system was executed and the result shows that this proposed system can help the personnel to maintain the database dynamically to provide the most up-to-date information, and at the same time, it allows the users to access the reliable and timely information in their use of the data. In other words, the updating and propagation of information can actually be beneficial to a real *DDBS*.

References

[1] B. Abramson, "ARCO1: An Application of Belief Networks to the Oil Market," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 1-8, 1991.

[2] J.M. Agosta, "The Structure of Bayes Networks for Visual Recognition," R. Shacter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence 4*, pp. 397-405, 1990.

[3] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper, "The ALARM Monitoring System: A Case Study With Two Probabilistic Inference Techniques for Belief Networks," *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pp. 247-256, 1989.

[4] C. Berzuini, R. Bellazzi, and D. Spiegelhalter, "Bayesian Networks Applied to Therapy Monitoring," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 35-43, 1991.

[5] D. Heckerman, J. Breese, and K. Rommelse, "Sequential Trouble-shooting Under Uncertainty," *Proceedings of the Fifth International Workshop on Principles of Diagnosis*, pp. 121-130, 1994.

[6] W. Lam, "Bayesian Network Refinement via Machine Learning Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, March 1998.

[7] S.A. Musman, L.W. Chang, and L.B. Booker, "Application of a Real-Time Control Strategy for Bayesian Belief Networks to Ship Classification Problem Solving," *International Journal, Pattern Recognition and Artificial Intelligence*, vol.7, no. 3, pp. 513-526, 1993.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Revised second printing, Morgan Kaufmann, 1988.

[9] J. Pearl, "Fusion, Propagation, and Structuring in Belief Networks," *Artificial Intelligence*, 29, pp. 241-288, 1986.

[10] Y. Peng and J.A. Reggia. *Abductive Inference Models for Diagnostic Problem-Solving*, Springer-Verlag, 1990.