Building Bayes Nets with Semistructured Probabilistic DBMS

Wenzhong Zhao, Alex Dekhtyar, Judy Goldsmith,
Erik Jessup and Jiangyu Li
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046
{wzhao0, dekhtyar, goldsmit, ejjess2 and jli2}
@cs.uky.edu

Abstract

Bayes nets appear in many Artificial Intelligence applications that model stochastic processes. Efficiently building Bayes nets is crucial to the applications. In this paper we describe our approach to building, updating and maintaining large Bayes net models. This approach is based on our implementation of the Semistructured Probabilistic Database Management System (SPDBMS) that provides us with robust storage and retrieval mechanisms for large quantities of probability distributions. On top of SPDBMS, we build client applications designed to deal with specific sub-tasks within the model construction problem. The two applications described here are the Bayes Net Builder (BNB) that allows knowledge engineers to describe the structure of the Bayes Net model, and the Probability Elicitation Tool (PET) designed to elicit conditional probability distributions from the domain experts.

1 Introduction

In recent years Bayes nets have been used in a wide array of AI applications from medical to military. Better planning and inference engines allow for larger and more complex stochastic domains to be modeled and processed. Presently, there are some robust commercial and open-source Bayes network inference software systems, such as [1, 3, 4]. All of them bring the process of network construction and inference directly to a single desktop by integrating the inference engine with the front end that allows the user to design the network structure and input all necessary conditional probability tables.

However, modeling large, dynamic domains often requires separation of tasks, rather than their integration. In particular, to design a complete Bayes Network model for a large application, a team of knowledge engineers must determine the random variables present in the application and their domains; possible meta-information attributes that describe specific situations in the application; conditional dependencies between the domain

elements (Bayes network structure). A team of domain experts specify the conditional probability tables associated with Bayes network nodes and specific situations. While determination of variables must precede all other tasks, determination of the network structure and construction of the conditional probability tables often become parallel, and even asynchronous processes.

The Bayes network development suite described in this paper addresses exactly this problem. It is built on the idea of a central repository that stores all the currently available information about network structure and conditional probability tables. This repository resides on a central database server and is updated and accessed by special-purpose software that can perform specific model-building subtasks. In our solution, the data repository is a Semistructured Probabilistic database, and its initial version is described in [2]. We have implemented a Semistructured Probabilistic DBMS server [5] that provides access to the stored data via the queries expressed in the Semistructured Probabilistic query algebra [6].

2 The Bayes Network Development Suite

2.1 Overall System Architecture

The Bayes network development suite described here consists of three components, all implemented in Java. The overall architecture is shown in Figure 1. The backbone of the system is the Semistructured Probabilistic DBMS (SPDBMS) server [5]. Two other components were developed. Bayes Net Builder (BNB) allows its users to define the application domain and construct the Bayes network structure for it. Probability Elicitation Tool (PET) facilitates elicitation of conditional probability tables from domain experts. Both PET and BNB use SPDBMS as the source of their input and the repository of the data they generate.

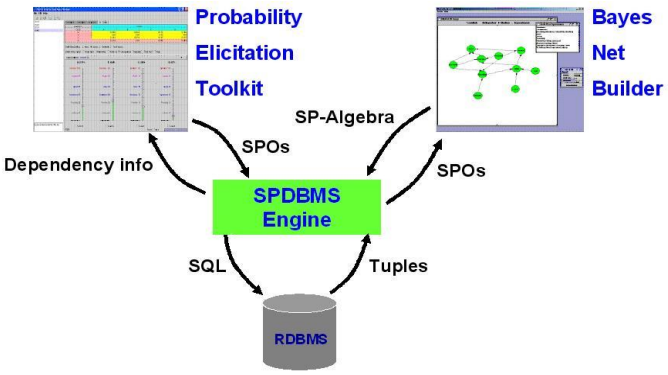


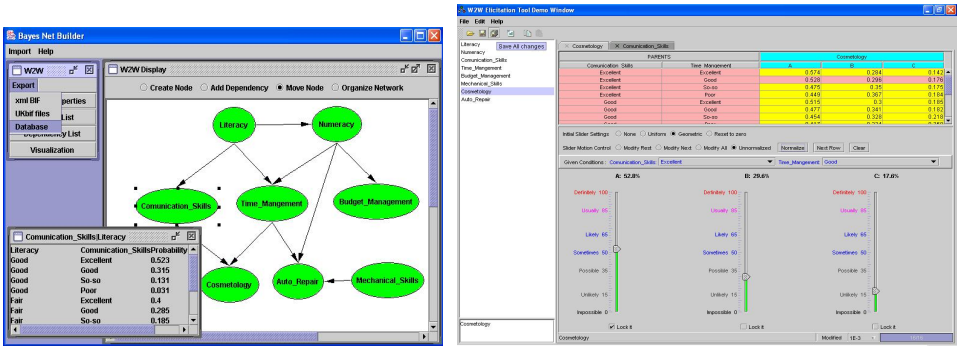
Figure 1: The overall system architecture.

To that extent, SPDBMS server allows storage and retrieval of two types of information: XML encoding of the Bayes network structure and SPOs. For the Bayes network

structure encodings, stored simply as files, storage and retrieval operations are implemented, with storage operation validating the XML file against the Bayes network structure XML Schema. To store SPOs, SPDBMS server uses as its back end a relational database accessed via an RDBMS (currently – Oracle 8i) For SPOs, the SPDBMS server implements the insertion and deletion operations, as well as the SP-Algebra query operations [6]. The server provides a convenient Java API through which BNB and PET (as well as other client applications in general) can connect to it, and pass information processing and retrieval instructions.

2.2 Model Construction

The process of Bayes network construction starts with the Bayes Net Builder. This tool (see Figure 2.(a) provides a simple GUI that allows knowledge engineers to specify random variables, associated domains and dependencies between the variables. The interface, similar to the graph-drawing interfaces of standard Bayes network inference tools such as Hugin [3] and JavaBayes [1], allows the users to create network nodes on the canvas, describe their domains, move them around and draw edges signifying dependencies. As work on the Bayes net structure proceeds, BNB maintains the internal representation of the current state of the Bayes network. Once a preliminary network is constructed, it can be exported into both our internal XML format and XMLBif (without the probability tables, yet). The network description is sent to the SPDBMS server.



(a) Bayes net builder (b) Probability elicitation toolkit

Figure 2: Screenshots of the Bayes net builder and probability elicitation toolkit.

The Bayes net building process is distributed and almost asynchronous. The network description can be imported from the SPDBMS server into the Probability Elicitation Tool. This description allows the tool to construct the list of CPTs that need to be elicited from domain experts. The main window of PET is shown in Figure 2.(b). The domain expert (with or without the help of a knowledge engineer) works on constructing one CPT at a time. The user interface provides a number of different means for entering probability values: a spreadsheet-like environment at the top of the page, the vertical slider bars at the bottom of the screen that can be used to provide both normalized and non-normalized user probability assessments (and controlled via a variety of interface widgets) and a number

of common preset distributions. The slider scales use both numeric probability values and verbal probability scales allowing the users uncomfortable with numbers to express their probability distributions in terms like "usually", "frequently", "unlikely", "probably" and such. Non-normalized distributions entered by the user are automatically normalized by PET. Once a user is satisfied with a specific CPT (which may still be incomplete), (s)he can save it to the database. The CPT gets converted into a collection of SPOs as described above and submitted to the the SPDBMS server for insertion into the appropriate database.

The moment the CPT data (in SPO format) is stored in the SP Database, it becomes available to the Bayes Net Builder. Each time a node X (with parents $Y_1, \ldots, Y_k$) of the Bayes network B is selected in the Bayes Net Builder, the SP-Algebra query

$$\sigma_{v=\{X\} \wedge Y_1 \ni dom(Y_1) \wedge \ldots \wedge Y_k \ni dom(Y_k)}(B)$$

(meaning "Select from SP-relation B all SPOs with participating random variable X and conditioned by random variables $Y_1, \ldots, Y_k$") is issued to the SPDBMS server. At each moment of time, the answer set to this query is exactly the set of SPOs for the CPT of node X available in the database. BNB receives the answer set from the SPDBMS server, parses it, and displays the current state of the requested CPT. Figure 2.(a) pictures the state of the Bayes Net Builder after the node Communication_Skills was selected by the user.

3 Conclusion and Future Work

The framework proposed here can help teams of researchers build large and complex Bayes networks in distributed and asynchronous fashion. This is the first implementation (to our knowledge) to incorporate probabilistic database technology into a solution of an Artificial Intelligence problem. The architecture is flexible and extensible: new client applications for performing other model-building tasks, such as data extraction from databases, can be seamlessly incorporated into it.

The main development thrust is towards making it suitable for KBMC applications by allowing for storage and retrieval of situation-specific information and reasoning designed to build situation-specific Bayes net models based on the stored data.

References

[1] Fabio Cozman. Bayes networks in java. http://www-2.cs.cmu.edu/ javabayes, 1998.

[2] Alex Dekhtyar, Judy Goldsmith, and Sean Hawkes. Semistructured probabilistic databases. In *Proc. Statistical and Scientific Database Management Systems*, 2001.

[3] Hugin. Hugin expert a/s. http://www.hugin.dk, 1998.

[4] Microsoft. Micrsoft belief network tools. http://www.research.microsoft.com/adapt/MSBNx, 1998.

[5] Wenzhong Zhao, Alex Dekhtyar, and Judy Goldsmith. A framework for management of semistructured proabilistic data. Technical Report TR385-03, Department of Computer Science, University of Kentucky, 2003.

[6] Wenzhong Zhao, Alex Dekhtyar, and Judy Goldsmith. Query algebra for interval probabilities. In *the 14th International Conference on Database and Expert Systems Applications, LNCS 2736*, pages 527 – 536, 2003.