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}@cs.uky.edu
ejjess2@uky.edu and jli2@engr.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 Bayes Net Builder (BNB) that allows knowledge engineers to describe the structure of the Bayes Net model, and Probability Elicitation Tool (PET) designed to elicit conditional probability distributions from domain experts.

1 Introduction

In recent years Bayes nets have been used in a variety 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 [Coz98, Hug98, Mic98]. 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, and 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, or even asynchronous processes.

The Bayes network development suite described in this paper addresses exactly this problem. In our solution, the data repository is a Semistructured Probabilistic database, described in [DGH01]. We have implemented a Semistructured Probabilistic DBMS server [ZDG03a] that provides access to the stored data via the queries expressed in the Semistructured Probabilistic query algebra [ZDG03b].

2 The Bayes Network Development Suite

2.1 Semistructured Probabilistic Objects (SPOs)

In this section, we briefly describe the Semistructured Probabilistic object data model. For more complete description of the model we refer the reader to [DGH01, ZDG03a].

Semistructured Probabilistic databases store a variety of probability distributions. Each individual probability distribution is stored as a single SPO. An SPO S is a tuple $\langle T, V, P, C\rangle$, where $V = \{v_1, \ldots v_k\}$ is the list of participating random variables of S; P is the probability table of S that associates probability values with instances from $dom(V) = dom(v_1) \times \ldots \times dom(v_k)$; C stores the conditional information for S; T is the context (otherwise known as meta-information) associated with S.

2.2 Overall System Architecture

The Bayes network development suite described here consists of three components, as shown in Figure 1. The backbone of the system is the Semistructured Probabilistic DBMS (SPDBMS) server [ZDG03a]. Two other components have been developed. Bayes Net Builder (BNB) allows its users to define the application domain and construct the Bayes network structure for it. The 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.

To that extent, the 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, encoded as an XML file, storage and retrieval operations are implemented with validation against the Bayes network structure XML Schema. For SPOs, the SPDBMS server implements the insertion and deletion operations, as well as the SP-Algebra query operations [ZDG03b]. 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.

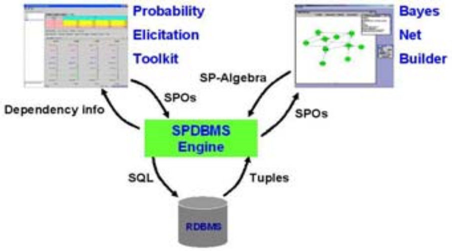


Figure 1: The overall system architecture.

2.3 Model Construction

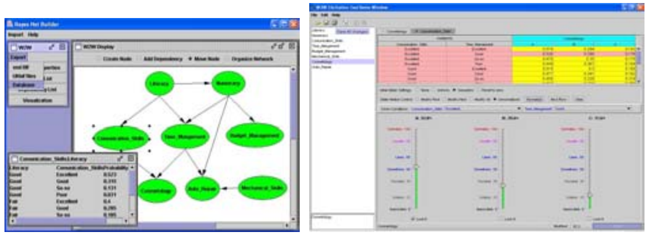
The process of Bayes network construction starts with the Bayes Net Builder. This tool (see Figure 2.(a)) provides a simple GUI that allows knowledge engineers to specify random variables, associated domains and dependencies between the variables. The interface, similar to the graph-drawing interfaces of standard Bayes network inference tools such as JavaBayes [Coz98] and Hugin [Hug98], allows 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. The network description is sent to the SPDBMS server.

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 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 input both normalized and unnormalized probability assessments and a number of common preset distributions. Unnormalized distributions entered by users 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 and submitted to 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, 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.



(a) Bayes net builder

(b) Probability elicitation toolkit

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

3 Conclusion

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.

Acknowledgements. This work was partially supported by NSF grant ITR-0325063.

References

[Coz98] Fabio Cozman. Bayes networks in java. http://www-2.cs.cmu.edu/ javabayes, 1998.

[DGH01] Alex Dekhtyar, Judy Goldsmith, and Sean Hawkes. Semistructured probabilistic databases. In *Proc. Statistical and Scientific Database Management Systems*, 2001.

[Hug98] Hugin. Hugin expert a/s. http://www.hugin.dk, 1998.

[Mic98] Microsoft. Micrsoft belief network tools. http://www.research.microsoft.com/adapt/MSBNx, 1998.

[ZDG03a] 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.

[ZDG03b] 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.