

A Web-based Interface for Hiding Bayesian Network Inference

C.J. Butz¹, P. Lingras², and K. Konkel¹

¹ Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {butz,konkel1k}@cs.uregina.ca

² Department of Math and Computing Science, Saint Mary's University
Halifax, NS, Canada B3H 3C3
E-mail: pawan.lingras@stmarys.ca

Abstract. *Bayesian networks* have been applied for several uncertainty management problems in the artificial intelligence and Web intelligence communities. However, one may require the use of Bayesian networks, yet lack the background knowledge to build them. Moreover, it is widely acknowledged in the Bayesian network community that understanding Bayesian network inference is an arduous task. In this paper, we solve this dilemma by proposing a Web-based interface for hiding Bayesian network inference. This approach allows a much wider audience to utilize Bayesian network inference without having to understand how the inference process is actually carried out.

1 Introduction

Bayesian networks (BNs) [11] are an established framework for uncertainty management in the artificial intelligence community. A BN consists of a *directed acyclic graph* (DAG) and a corresponding set of *conditional probability tables* (CPTs). The *probabilistic conditional independencies* (CIs) [13] encoded in the DAG indicate that the product of CPTs is a joint probability distribution. Exact inference algorithms in BNs can be broadly classified into two categories. One approach is *join tree propagation*, which systematically passes messages in a join tree constructed from the DAG of a BN (see [8] for example). The second approach to BN inference is *direct computation*, which performs inference directly in a BN. Very recently, Madsen [7] examined hybrid approaches to BN inference. Of the three hybrid approaches tested, *Lazy arc-reversal* (Lazy-AR) was empirically shown to be the state-of-the-art method for exact inference in discrete BNs [7]. In [3], we proposed *Lazy arc-reversal with variable elimination* (Lazy-ARVE) as a new approach to BN inference and illustrated its benefits over Lazy-AR.

It is a difficult challenge, however, to learn how BN inference algorithms work [5, 6, 12]. Jensen [6], one of the founders of BNs, explicitly states that probabilistic reasoning literature is not meant for readers looking for a way into

the field. In fact, Russell and Norvig [12] state that some of the mathematics and notation are unavoidably intricate even for inference in singly connected BNs [11]. Since the ideas and techniques of BNs are rather complex, they have not spread much beyond the research community responsible for them [5]. The above remarks provide clear motivation for the work now presented.

In this paper, we suggest the use of a Web-based interface which allows users to utilize the powerful Bayesian network framework without having prior experience in the area. More specifically, a user is able to use the Web for uploading a data set that will be used in reasoning, as well as for posing queries about the supplied problem domain. Our system will learn a BN from the supplied data set, and, more importantly, it will perform BN inference on behalf of the user. By serving as a buffer between the inexperienced user and the sophisticated BN inference techniques, the primary advantage of our system is that it allows a much larger audience to apply BNs in practice. The work here corresponds to the notions of *three-level architecture* in databases and *abstraction* in computer programming languages, both of which are central to their fields. While it is acknowledged that no new technical contribution is made, the advantage of this work is that it allows non-experts to take full advantage of the proven BN technology.

This paper is organized as follows. In Section 2, Bayesian networks are reviewed. In Section 3, the architecture of our prototype interface is outlined. Related works are discussed in Section 4. The conclusion is given in Section 5.

2 Bayesian Networks

Let U be a finite set of discrete random variables, each with a finite set of mutually exclusive states. It may be impractical to define a joint distribution on U directly: for example, one would have to specify 2^n entries for a distribution over n binary variables. BNs utilize *conditional independencies* [13] to facilitate the acquisition of probabilistic knowledge.

Let X, Y and Z be disjoint subsets of variables in R . Further, let x, y , and z denote arbitrary values of X, Y and Z , respectively. We say Y and Z are *conditionally independent* given X under the joint probability distribution p , denoted $I(Y, X, Z)$, if $p(y|x, z) = p(y|x)$, whenever $p(x, z) > 0$.

A *Bayesian network* (BN) [11] is a pair $\mathcal{B} = (D, C)$. In this pair, D is a *directed acyclic graph* (DAG) on a set U of variables, and $C = \{p(a_i|P_i) \mid a_i \in D\}$ is the corresponding set of *conditional probability tables* (CPTs), where P_i denotes the *parent set* of variable a_i in the DAG D . We will use the terms BN and DAG interchangeably if no confusion arises. For example, consider the BN $\mathcal{B} = (D, C)$, where D is the DAG in Fig. 1 on $U = \{a, b, c, d, e, f, g, h, i, j, k\}$, and C is the corresponding set of CPTs. The conditional independencies encoded in the DAG D indicate that the product of the CPTs in C define a *unique* joint probability distribution $p(U)$:

$$p(U) = p(a)p(b)p(c|a)p(d|b)p(e|b)p(f|d, e)p(g|b) \\ p(h|c, f)p(i|g)p(j|g, h, i)p(k|h). \quad (1)$$

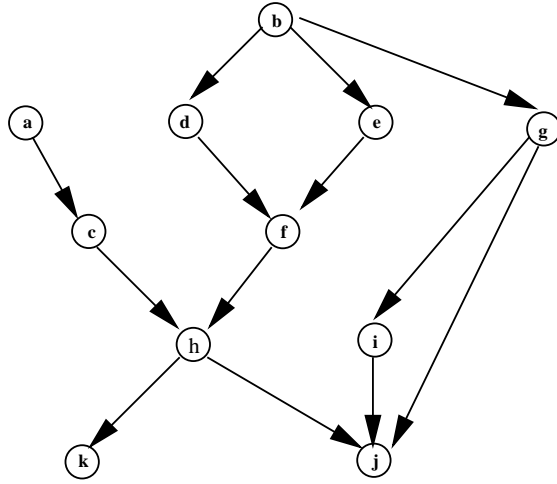


Fig. 1. A Bayesian network.

3 A Prototype Interface

We begin this section by outlining the main components of our Web-based interface for hiding Bayesian network inference and end it by showing some screenshots of our implementation.

The user of our system will provide two kinds of input, namely, the sample data to be reasoned with, and queries posed to the system. We assume that the former will be given once, while the latter can be posed multiple times interactively. The data is presented in tabular form and the domain of each attribute (variable) is defined as the set of values appearing for this attribute in the sample data. Our system provides a menu box where the user can upload the data onto our system. Once the sample data resides on our computer system, we apply *Netica* [10] to learn a BN from this data. At this point, we are ready for the second kind of input from the user.

Our system provides a screen in which the user can enter queries. In other words, the user can request $p(X|Y = y)$, where the collected evidence is $Y = y$ and the user is interested in the set X of target variables. For example, in a Web-based intelligent system like [4], the system may be asked for the probability of a student “passing the exam” (X), given that the student “failed the midterm examination” ($Y_1 = y_1$) and “has not completed the sample exercise questions” ($Y_2 = y_2$).

Given an input query $p(X|Y = y)$, our system computes the posterior probabilities of the variables in X using our own inference algorithm, which is based on the recent work in [3]. More specifically, our algorithm is a hybrid approach utilizing join tree propagation to guide the inference procedure. However, direct

computation techniques are taken advantage of to perform the physical computation on the stored probability distributions. Moreover, our approach performs parallel computation whenever possible in order to determine the requested probabilities as soon as possible.

Once $p(X|Y = y)$ has been physically computed, our system displays the posterior probabilities for the variables in X on a Web page for the user. Based on the calculated probabilities, the user can make required decisions and take any appropriate action. This may include posing further queries to our system.

The important point is that our Web-based interface allows BNs to be utilized by the user without the user having to understand the inference process in BNs. Our system serves as a shield between the user and BN inference, thereby sparing the user the arduous task of learning how BNs work (see Section 3).

We conclude this section by demonstrating an example session of our system, which is currently only implemented as a browser-based system. The welcoming screen-shot is given in Fig. 2. As indicated, the user can either upload a data set or request posterior probabilities regarding the supplied data. Thus, the user will first upload a data set. Our system will then learn a BN from the given data. Let us assume that the BN in Fig. 1 is obtained from the supplied data. The BN CPTs obtained from the data set are not depicted as they are not pertinent to our discussion. At this point, our system is ready to perform BN inference.

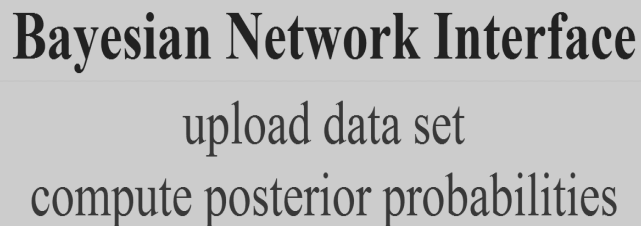


Fig. 2. Welcoming page of our interface.

The user is able to set observed values based on the evidence collected in their own application domain. Fig. 3 shows the user setting the values of variables f and h , namely, $f = 0$ and $h = 0$. Given the entered evidence, our system performs reasoning in the BN of Fig. 1 on behalf of the client and provides the posterior probabilities of the other variables, as depicted in Fig. 4. The user can make the appropriate decisions based on these returned probabilities and act accordingly.

4 Related Works

In this section, we briefly contrast our interface to Web-based BN inference with some related works.

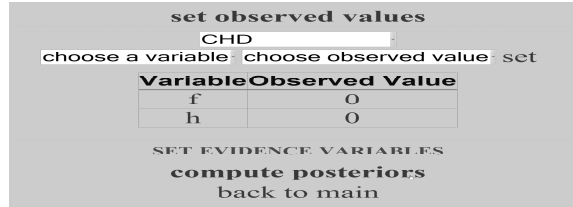


Fig. 3. Entering collected evidence $f = 0$ and $h = 0$.

Posterior Probabilities for CHD BN									
Variable Observed Value									
f									
h									
11 VARIABLES IN THE CHD TABLE									
ap(a f=0,h=0)	bp(b f=0,h=0)	cp(c f=0,h=0)	dp(d f=0,h=0)	ep(e f=0,h=0)	fp(f f=0,h=0)	gp(g f=0,h=0)	hp(h f=0,h=0)	ip(i f=0,h=0)	jp(j f=0,h=0)
0	0.50	0	0.91	0	0.42	0	0.50	0	0.50
1	0.50	1	0.09	1	0.58	1	0.50	1	0.50
0	1.00	0	0.91	0	1.00	0	0.64	0	0.62
1	0.00	1	0.09	1	0.00	1	0.36	1	0.38
Kp(k f=0,h=0)	0	0.50							
1	0.50								

Fig. 4. Our system returns the posterior probabilities of all other variables in the BN of Fig. 1 given the collected evidence in Fig. 3.

Myllymki et al. [9] proposed a Web-based BN system to demonstrate how to build a dependency model out of a data set, and how to use it for finding interesting relations in the data. One aim of this system is to teach the user how to perform Bayesian dependency modeling and Bayesian inference. In contrast, our Web-based BN system seeks to hide these modeling and inference tasks from the user.

A Web-based BN viewer, called *BayesNet*, was suggested in [1]. Similar to the above system, BayesNet's objective is to allow viewing and working with BNs. On the contrary, we assume that the user lacks the knowledge to use BNs but still wants to take advantage of them.

It is worth mentioning that the *Uncertainty in Artificial Intelligence* (UAI) community has investigated a standard XML format for BNs, called *Bayesian networks Interchange Format* (BNIF) [2]. The stated goal of the BNIF discussions is to make it easy for UAI researchers using different BN modeling and

inference tools to share models with ease. Once again, our purpose is very different - it is to allow researchers with limited BN knowledge the ability to take advantage of BNs in their Web applications.

5 Conclusion

This is the first work to suggest a Web-based interface for hiding BN inference. Our interface allows researchers and practitioners access to the established BN framework without having prior expertise in the area. It can then be seen that the work here corresponds to the notions of *three-level architecture* in databases and *abstraction* in computer programming languages, both of which are central to their fields. We have contrasted our work with other related works of similar names by emphasizing opposing objectives. Whereas these other systems use the Web to teach users the intrinsics of Bayesian network inference, our Web-based interface instead tries to shield the user from these complicated issues. While it is acknowledged that no new technical contribution is made, the work here allows any Web user to exploit BNs. We have implemented a prototype interface and have included screen shots illustrating its functionality.

References

1. <http://www.webbayes.net/>.
2. <http://research.microsoft.com/dtas/bnformat/>.
3. C.J. Butz and S. Hua, An Improved Lazy-AR Approach to Bayesian network Inference, *Nineteenth Canadian Conference on Artificial Intelligence*, 183-194, 2006.
4. C.J. Butz, S. Hua and R.B. Maguire, A Web-based Intelligent Tutoring System for Computer Programming, *IEEE/WIC/ACM International Conference on Web Intelligence*, 159-165, 2004.
5. E. Charniak, Bayesian networks without tears, *The AI Magazine*, **12** (4), 50-63, 1991.
6. F.V. Jensen, *An Introduction to Bayesian Networks*, UCL Press, London, 1996.
7. A.L. Madsen, An empirical evaluation of possible variations of lazy propagation, *Proc. 20th Conference on Uncertainty in Artificial Intelligence*, Banff, Canada, 366-373, 2004.
8. A.L. Madsen, F.V. Jensen, Lazy propagation: A junction tree inference algorithm based on lazy evaluation, *Artif. Intell.*, **113** (1-2), 203-245, 1999.
9. P. Myllymki, T. Silander, H. Tirri, and P. Uronen, B-course: A web-based tool for Bayesian and causal data analysis, *International Journal on Artificial Intelligence Tools*, **11** (3), 369-387, 2002.
10. Netica. Norsys: Software corp. <http://www.norsys.com/netica.html>, 2000.
11. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1998.
12. S.J. Russell, and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, 2003.
13. S.K.M. Wong, C.J. Butz and D. Wu, On the implication problem for probabilistic conditional independency, *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans, **30** (6), 785-805, 2000.