Modeling First-Order Bayesian Networks (FOBN)

A Comparative Study of BLOG, BLP and MEBN

Saleha Raza
Artificial Intelligence Lab, Faculty of Computer Science
Institute of Business Administration
Karachi, Pakistan
saleha.raza@khi.iba.edu.pk

Sajjad Haider
Artificial Intelligence Lab, Faculty of Computer Science
Institute of Business Administration
Karachi, Pakistan
sajjad.haider @khi.iba.edu.pk

Abstract— **Bayesian networks provide an elegant formalism to perform inferences under uncertainty. Their shortcoming of being propositional in nature, however, restricts their expressive power and restrains their use in domains where number of instances may vary from situation to situation. First-order Logic (FOL), on the other hand, enjoys that power of expressiveness but is deterministic in nature. Integration of Bayesian networks and first-order logic provides powerful mechanism to capture and process domains that are truly dynamic and non-deterministic. The paper explores and compares three different probabilistic languages, namely Bayesian Logic Program (BLP), Bayesian Logic (BLOG) and Multi-Entity Bayesian Network (MEBN) that provide support to develop First Order Bayesian Networks (FOBN). The study identifies key characteristics that are prevalent in all three languages and compares their relative strengths and weaknesses.**

Keywords: first-order Bayesian network, probabilistic languages, MEBN, BLP, BLOG

I. INTRODUCTION

The innate ability of Bayesian networks (BN) [1] to capture uncertainty has made them widely applicable in real-life domains that are non-deterministic in nature [2,3]. Bayesian networks, however, are propositional in nature and represent particular instances of a domain rather than its generalized concept. This behavior limits their power of expression and hinders their use in variety of situations. First-order Logic (FOL) [4] gives generalized representation of objects or domains and hence enjoys that power of expression that Bayesian networks lack. They, however, offer very limited support to handle uncertainty and hence lack in areas where Bayesian networks excel. The integration of Bayesian networks and first-order logic can provide a powerful mechanism to capture and reason in domains that are truly dynamic and non-deterministic. Several efforts have been made in this dimension and different models/languages have been proposed [4-8].

The purpose of this study is to provide an insight into requirements of first-order Bayesian networks by exploring and experimenting with languages that support this integration. Three languages discussed in this paper are Bayesian Logic Program (BLP) [5], Multi-Entity Bayesian Networks (MEBN)[4,9] and Bayesian Logic (BLOG) [6,10]. Some Some studies [10-12] have been done to analyze few of these languages, but they either do not cover MEBN or provide a survey report without comparing the modeling strengths and weaknesses of these languages and the results produced by them. The key contribution of this paper is to present a comparison based on implementation in the three selected languages. The effort can serve as a good starting point for people (both researchers and practitioners) who are about to step into the area of first-order Bayesian networks.

The comparison, in this paper, has been done from the view point of a knowledge engineer as she models a situation using FOBN. A simple scenario is implemented in all three languages and the implementations are then compared on the bases of modeling intuitiveness, expressive power, modularity and consistency. The comparison of inference/learning mechanisms and their complexity is not within the scope of this paper.

The overall organization of this paper is as follows. Section 2 describes a sample scenario and highlights limitations of BN and FOL in modeling it. Section 3 gives an overview of BLP, MEBN and BLOG and how the sample scenario is implemented in each language. Based on these implementations, Section 4 performs a comparison and analyzes the strengths and weaknesses of these languages. Finally, Section 5 concludes the paper and provides future research directions in the field of first-order Bayesian networks.

II. SAMPLE SCENARIO

This whole study is based on a book-rating scenario given below:

"Overall sale of a newly published book is highly influenced by the rating of its author(s). Author rating, in turn, is dependent on how good or bad the readers have rated his previous books"

Fig. 1 gives few variants of Bayesian networks that model this scenario. Fig. 1(a) models a simple situation in which a book has been written by an author who has previously written just one book. Fig. 1(b) models a situation where a book is written by an author who has already written three other books. Finally, Fig. 1(c) represents a situation in which a book has two authors and both of them have already authored different books. Following this pattern, there can be many combinations for varying number of books and authors, and for each distinct combination a separate structure of Bayesian network will

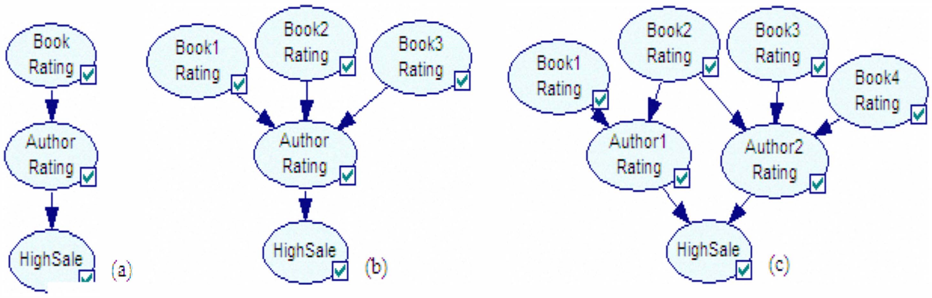

Figure 1 - Different instances of Bayesian network for book-rating scenario

be formed. These cases highlight the propositional nature of Bayesian network representing particular instances of real-world objects, rather than their generalized concept.

The first-order logic based representation of the same scenario is given below:

Book(X); Author(Y); WrittenBy(X,Y)

AuthorRating(A)⟵ WrittenBy(B,A), BookRating(B)

HighSale(B) ⟵ WrittenBy(B,A), AuthorRating(A)

FOL, though gives true logical representation of the problem in hand, is unable to model probabilistic relationships among different clauses.

III. FIRST ORDER BAYESIAN NETWORK (FOBN)

First-Order Bayesian Networks (FOBN) is an integration of Bayesian networks and first-order logic that aims to exploit probabilistic support of Bayesian networks and expressive power of first-order logic. Different languages have been developed to support this integration. This section gives an overview of three languages, namely BLOG, BLP and MEBN, and presents a sample scenario implemented in these languages.

A. Bayesian Logic Program (BLP)

Bayesian Logic Program (BLP) [5] is a refinement of three other probabilistic languages, namely probabilistic logic program[13], relational Bayesian networks[14] and probabilistic relational model[15] and provides a common kernel for them. A key design principle of BLP states that "*while combining first order logic with Bayesian nets, the resulting formalism should be as close as possible to both Bayesian nets and to some well-founded first order logic knowledge representation mechanism* [5]". Balios[1] is an inference engine for BLP.

Bayesian logic program is a sequence of two types of clauses: logical and Bayesian. Logical clauses are mapped to predicates in logic programs and are either true or false, while Bayesian clauses represent probabilistic concepts or relationships. Strength of probabilistic influence is specified by associating Conditional Probability Table (CPT) to each bayesian clause. Fig. 2 (a) is BLP of book-rating scenario, implemented in Balios, in which filled and unfilled ovals represent bayesian & logical clauses respectively and squares denote probabilistic relationships among nodes. While running query, a target node is specified and support network, as shown in Fig. 2 (b), is generated. In case of having multiple clauses for same bayesian term, BLP provides mechanism to define *combining rules* and *aggregations that* enable modeler to construct one CPT from several CPTs available.

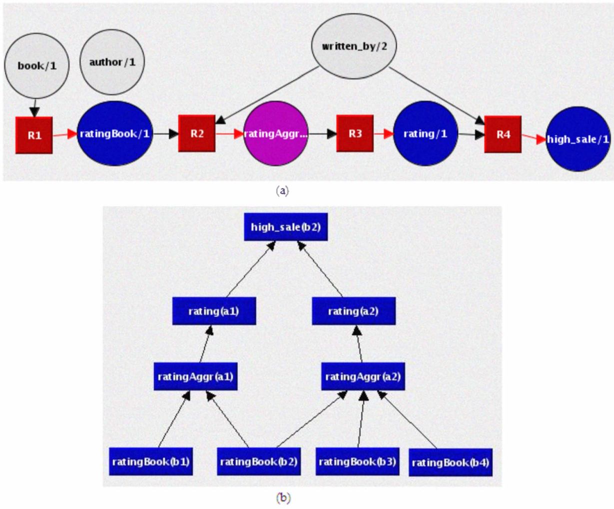

Figure 2 – (a) BLP model and (b) its SSBN

B. Multi-Entity Bayesian Network (MEBN)

MEBN provides a framework to give logic based representation of Bayesian networks. Each MEBN model, called *MTheory*, represents a particular domain of discourse. Different subjects of that domain are represented by smaller components known as *MFrag*. MFrag provides grouping of entities and their characteristics pertinent to the respective subject. Each MFrag contains a DAG with parameterized nodes representing attributes of entity and edges representing dependencies amongst them. MFrag node can be of three types: *Context Nodes* are evaluated to either true/false when substituted with constant values in place of

[1] http://people.csail.mit.edu/kersting/profile/balios/BALIOS_download.html

parameters. *Resident Nodes* are local nodes of MFrag that represent random variables and have their probability distributions. *Input Nodes* serve as input to derive probability distribution of resident nodes. They are resident nodes of other MFrag where their own probability distribution is defined. Fig. 3 gives MTheory of our sample scenario, implemented in UnBBayes[2], a tool that supports implementation of MEBN model. Here, ovals represent resident nodes while trapezoid and pentagons represent input and context nodes respectively.

With the help of context nodes & local probability distribution, MEBN provides mechanism to quantify (either existentially or universally) over all parents of a given node. MTheory, which is a coherent collection of MFrags, represents a problem domain. Specific instances of this domain are created by substituting parameters with explicit values obtained from situation in hand. This results in construction of *Situation Specific Bayesian Networks (SSBN)* that depicts a particular occurrence of that problem. SSBN, generated this way, is a standard Bayesian network and all standard inference mechanisms of Bayesian network are applicable on SSBN.

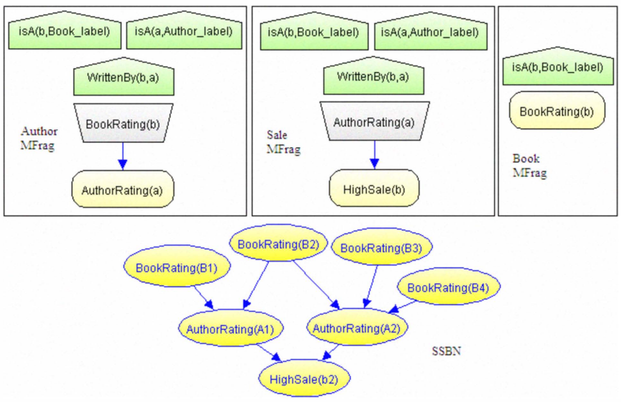

Figure 3 - MEBN model and its SSBN

C. Bayesian Logic (BLOG)

Bayesian Logic (BLOG) [6] is another probabilistic language that supports representation of first-order Bayesian networks. BLOG further enriches this concept by providing support for unknown objects, objects that exist in domain but are not pre-specified and are known once they are encountered. This study focuses on domains with known objects and, hence explores the subset of BLOG engine[3] that is sufficient to model first-order Bayesian networks. Table 1 shows code snippet of book-rating model in BLOG.

Table 1 - Code snippet of BLOG for book-rating scenario

```
type Book; type Author; type RatingLevel;
guaranteed RatingLevel Low, Medium, High;
nonrandom Boolean WrittenBy(Book, Author);
random RatingLevel BookRating(Book);
random RatingLevel AuthorRating(Author);
random Boolean HighSale(Book);

BookRating(b) ~ TabularCPD[[0.3,0.6,0.1]];
AuthorRating(a)
    {if (exists Book b(WrittenBy(b,a)
                      & BookRating(b)=High))
    Then ~ TabularCPD[[0.1,0.2,0.7]]
elseif !(exists Book b (WrittenBy(b,a)
                    & BookRating(b)=Low))
Then ~ TabularCPD[[0.15,0.7,0.15]]
Else ~ TabularCPD[[0.3,0.6,0.1]] };

//Known World
guaranteed Author a1,a2,a3,a4;
nonrandom Boolean WrittenBy(Book, Author)
  = ListInterp[a2, b1, a1, b2, a1, b2,a2,
 b3, a2, b4, 2,];
//Evidence
obs BookRaging(b1) = High;
//Query
query HighSale(b2)
```

'*type*' statements define different types that exist in domain of discourse. '*guaranteed*' statement specifies those instances of particular type that are not situation specific and are guaranteed to exist in all instances of given model. '*nonrandom*' and '*random*' functions represent deterministic and probabilistic clauses, respectively and parameterization of these functions provides ability to create multiple instances of these clauses.. Each random function has its respective CPT. Probabilistic dependencies of one random function on others are captured by *dependency statements*. '*obs*' and '*query*' statements are meant to provide evidences and make query. When defining dependencies, it is not possible to specify static CPT for a random function that is dependent on unbounded number of instances of other random functions. In these situations, BLOG provides sophisticated support for universal and existential quantification.

IV. Comparison

This section presents comparison of the three languages based on the respective implementation of book-rating scenario. The comparison has been done in two parts. The first part compares probability results obtained through the three selected languages and validates them against the probabilities obtained through GeNie[4] for an equivalent standard Bayesian network. The second part analyses modeling support of these languages and uses model intuitiveness, expressive power, consistency, and integrity

[2] http://sourceforge.net/projects/unbbayes/

[3] http://people.csail.mit.edu/milch/blog/software.html

[4] http://genie.sis.pitt.edu

as key metrics to assess their strengths and potential weaknesses.

A. *Query Results Comparison*

The following test scenario is used to perform probability comparison.

"There are two authors, a1 and a2. Book b1 has been written by a1 while books b3 and b4 have been written by a2. Book b2 has been jointly written by both the authors."

Table 2 gives marginal probabilities of HighSale (HS) for each book and Table 3 shows posterior probabilities after providing evidence that BookRating(b1) is high. The probability results closely match with each other and with GeNie, and thus imply a consistent inference mechanism across all three languages. Slight variations in probabilities are mainly because of the simulation based inference mechanism in couple of languages.

Table 2- Marginal probabilities for HighSale (HS)

	HS (b1)	HS (b2)	HS(b3)	HS(b4)
GeNie	0.53	0.55	0.54	0.54
BLP	0.53	0.55	0.54	0.55
BLOG	0.52	0.55	0.54	0.54
UnBBayes	0.53	0.55	0.54	0.54

Table 3 - Posterior probabilities for HighSale (HS)

	HS*(b1)	HS*(b2)	HS*(b3	HS*(b4)
GeNie	0.66	0.67	0.54	0.54
BLP	0.62	0.64	0.51	0.56
BLOG	0.65	0.66	0.54	0.54
UnBBayes	0.66	0.67	0.54	0.54

B. *Modeling Features Comparison*

Preliminary features of FOBN: All three languages satisfactorily fulfill the core requirements of FOBN. This includes capability to handle variable number of instances, support for random and non-random clauses and relationships among them, specification of prior and conditional probabilities and support for recursion which gives them the ability to create dynamic Bayesian networks.

Intuitiveness: As far as intuitiveness is considered, BLP was found to be the most intuitive in a sense that it gives pretty natural extension of Bayesian networks by parameterizing nodes and supporting parameter binding. Moreover, it gives an equivalent rule-based representation of these networks. In addition, the specification of pre-defined combining rules and aggregations are far easier in BLP than in other languages. On the other hand, programming like syntax of BLOG may not be a preferred choice for many communities, but those with hands-on programming experience can easily adapt and find it more powerful than the others. MEBN provides better maintainability/modularity by grouping logically related nodes in MFrag – a feature not supported by other models. The feature, however, at times makes it non-intuitive for a knowledge engineer to come up with an initial MEBN model.

Expressive Power: Both MEBN and BLOG were found more expressive than BLP in modeling not-so-structured domains having nonlinearity in their dependencies. Both of them provide support for *quantification* (universal and existential) and *customized logic structures* for probability distribution. *Combining rules* are not directly available either in MEBN or in BLOG, but their equivalent logic can be written in both languages and is more customizable than predefined combining rules in BLP. MEBN additionally has support to define probabilities based on *influence count* which is the cardinality of influencing parent configurations. Both BLP and MEBN provide support for pre-defined *aggregation* functions. This support is not directly available in BLOG but workarounds are available in the form of customized java classes that can be plugged into the BLOG code.

Consistency Checking: typed nature of BLOG and MEBN makes it easy for them to assure consistency of model. BLP is not typed language and hence requires knowledge engineer to use logical clauses throughout model to ensure its correctness.

Modularity: With the use of MFrag, MEBN approach is modular and creates clusters of conceptually related nodes and capture uncertainty at its natural level of granularity. Such modularity, however, seems missing in the other two languages.

Programming Extensions: Being coupled with Prolog, BLP provides flexibility to write prolog statements for its logical predicates. BLOG exposes certain interfaces and classes that can be extended in java to enhance existing object model of BLOG. This gives ability to exploit full programming support of java. By the time of doing this study, MEBN has not exposed such support, but latest updates suggest that programming extensions are also being available in MEBN.

Specialized Features: There are certain specialized dimensions of these languages that are not pertinent to the sample scenario but are worth mentioning. BLOG works on open-world assumption and provides support for unknown objects - objects not known specifically but their existence in the modeled problem domain is known. BLOG engine gives support to query on set of objects, i.e., second-order language. MEBN theory enriches concept of capturing uncertainty by handling notions of Type uncertainty, Reference uncertainty and Existence uncertainty.

V. CONCLUSION

This study concludes that the focus of selected languages has primarily been on the qualitative/structural aspects of BN that gives them ability to create multiple instances and support for unbounded number of parents. All three languages satisfactorily fulfill these requirements for the sample scenario. Probability results obtained from all of them closely match and that implies consistency in underlying inference mechanism of these languages. But, we envisage that this integration urges significant enhancements in quantitative component of BN as well, the area that is yet to be refined. In experiment with three languages, MEBN relatively offers maximum support in defining probability distributions for its resident nodes. As we move forward with logic based representation of BN, Static CPTs and Noisy-OR combination may not be sufficient choice to model every situation and a natural desire would emerge to have extensive flexibility in defining probability distributions. This flexibility can be in the form of having variety of combining rules, aggregation, quantification, manipulation over cardinality of neighboring nodes, and support for customized logic building for probability distribution. Though, that may offer additional challenges to knowledge engineers.

ACKNOWLEDGMENT

The authors are greatly thankful to Prof. Kathryn Laskey and Rommel Carvalho of George Mason University (Decision Theory Group) and Brian Milch of MIT (CSAIL) for their valuable support during this study.

REFERENCES

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

[2] F.V. Jensen and T.D. Nielsen, *Bayesian Networks and Decision Graphs*, Springer, 2007.

[3] A. Mittal, *Bayesian Network Technologies: Applications and Graphical Models*, IGI Publishing, 2007.

[4] O. Pourret, P. Naïm, and B. Marcot, *Bayesian Networks: A Practical Guide to Applications*, Wiley, 2008.

[5] K.B. Laskey, "MEBN: A language for first-order Bayesian knowledge bases," *Artificial Intelligence*, vol. 172, Feb. 2008, pp. 140-178.

[6] K. Kersting and L.D. Raedt, *Bayesian Logic Programs*, Albert-Ludwigs University at Freiburg, 2001.

[7] B.C. Milch, "Probabilistic models with unknown objects," PhD Dissertation, University of California at Berkeley, 2006.

[8] Koller, D., & Pfeffer, A. Object-Oriented Bayesian Networks. Paper presented at the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97). San Francisco, CA, USA, 1997

[9] T. Sato, "A glimpse of symbolic-statistical modeling by PRISM," *J. Intell. Inf. Syst.*, vol. 31, 2008, pp. 161-176.

[10] David Heckerman, Christopher Meek, and Daphne Koller, "Probabilistic Entity-Relationship Models, PRMs and Plate Models, 2007.

[11] K. Kersting and U. Dick, "Balios: the engine for Bayesian logic programs," *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, Italy: Springer-Verlag New York, Inc., 2004, pp. 549-551.

[12] K. Kersting and L.D. Raedt, *Bayesian Logic Programming: Theory & Tool*, Albert-Ludwigs University at Freiburg, 2004 *Discovery in Databases*, Pisa, Italy: Springer-Verlag New York, Inc., 2004, pp. 549-551.

[13] P.C.G.D. Costa, "Bayesian semantics for the semantic web," PhD Dissertation, George Mason University, 2005.

[14] B. Milch and S. Russell, "First-Order Probabilistic Languages: Into the Unknown," *Inductive Logic Programming: 16th International Conference, ILP 2006, Santiago de Compostela, Spain, August 24-27, 2006, Revised Selected Papers*, Springer-Verlag, 2007, pp. 10-24.

[15] R. Braz, E. Amir, and D. Roth, "A Survey of First-Order Probabilistic Models," *Innovations in Bayesian Networks*, 2008, pp. 289-317.

[16] Bruynooghe et al., "An exercise with statistical relational learning systems," *International Workshop on Statistical Relational Learning, Belgium*, Jul. 2009.

[17] L. Ngo and P. Haddawy, "Probabilistic Logic Programming and Bayesian Networks," *Proceedings of the 1995 Asian Computing Science Conference on Algorithms, Concurrency and Knowledge*, Springer-Verlag, 1995, pp. 286-300.

[18] M. Jaeger, "Probabilistic-Logic Models: Reasoning and Learning with Relational Structures," *Proceeding of the 2008 conference on Tenth Scandinavian Conference on Artificial Intelligence: SCAI 2008*, IOS Press, 2008, pp. 197-200.

[19] D. Koller, "Probabilistic Relational Models," *Inductive Logic Programming*, 1999, pp. 3-13 Vangansewinkel, Tine; Van Hove, Lucie; Vennekens, Joost; Weytjens, Timmy; De Raedt, Luc., "An exercise with statistical relational learning systems," *International Workshop on Statistical Relational Learning, Belgium*, Jul. 2009