Grid-Enabled Bayesian Network

Yi-feng Zeng   Guo-quan Liu   Kim-leng Poh

Department of Industrial & Systems Engineering
National University of Singapore
Kent Ridge, Singapore, 117575
{g0202313, engp1518, isepohkl}@nus.edu.sg

Abstract

Bayesian network has been a successful tool in the decision support systems. In the changing world, the decision making demands adaptive Bayesian methods that are composed of Bayesian inferential and learning approaches. To achieve this goal, we propose a kind of grid-enabled Bayesian networks that intend to gridify Bayesian inferential and learning methods when the advanced grid computing techniques are integrated. Most of our effort is put into the discussion of grid-enabled learning methods and grid-enabled inferential methods as well as their challenging work on the integration. It is argued that grid-enabled Bayesian networks are able to utilize all available resources to support the adaptive decision making in the changing world.

Introduction

Bayesian network has been one of the most important ingredients in decision support systems that cope with uncertain issues in practice. In recent years, the concept of Bayesian artificial intelligence (Korb & Nicholson 2003) has been promoted with the aim at furthering the understanding of the nature of intelligence as well as producing useful tools for addressing difficult intellectual tasks. Its product must be intelligent, adaptive and reliable, and has the same performance as, if not better than humans.

To achieve this goal, one of the challenging works is to integrate Bayesian methods with computing technologies to build a grid-enabled Bayesian network. This work could be consummated with a marriage of Bayesian methods and computing technologies although both have long been studied in different disciplines. The Bayesian methods, including Bayesian reasoning and learning approaches, have been studied in the field of decision sciences for a long time. A lot of algorithms have been proposed, such as junction tree methods (Jensen et al. 1990) for Bayesian inference and block learning algorithms (Zeng & Poh 2004) for Bayesian learning. However, the huge computation cost of these algorithms is deferring their applications in the real world. On the other hand, grid computing (Foster et al. 2002) has been becoming very popular and it connects multiple regional and national computational resources to create a universal source of computing power. This powerful technology facilitates the computationally expensive applications by making use of distributed resources. However, to fully take advantages of grid computing, it requires that the problems to be solved be gridified.

An insightful investigation on these two issues will discover potential grid-enabled Bayesian network applications that are motivated for reducing the computation time. Hence, an integration of Bayesian methods and grid computing technology will allow users to utilize enormous computation resources to address complex and intellectual tasks, and then promptly provide results to support decision making in the changing world. However, the challenging work on their integration requires more effort. The following sections will specify some advanced Bayesian methods with the linkage of grid computing. We will investigate two things: the motivation for integration and the challenges involved in integrating them. In the discussion of grid-enabled Bayesian learning methods, we will generalize the bock learning algorithm in our recent work (Zeng & Poh 2005) and present its main procedures in detailed. After that, we intend to gridify the block learning algorithm as well as to discussion this challenging work. In the grid-enable Bayesian inferential methods, we argue that the junction tree inference algorithm, considered in an essential way, easily involves in the integration with grid computing techniques. Some similar work on this topic is valued.

This paper is organized as follows. In Section 2, we investigate grid-enabled Bayesian learning methods. As an important element associated with these methods, the block learning algorithm is generalized and discussed in detail as well as the challenging work concerning its integration with grid. In Section 3, we discuss grid-enabled Bayesian inferential methods. Finally, in Section 4, we conclude the paper.

Grid-Enabled Bayesian Learning Methods

An adaptive algorithm for learning Bayesian networks is a pillar in building Bayesian tools. It may differ from other algorithms in some abilities: (1) It could be scaled up; (2) It could be configured for tasks; (3) It could embrace other algorithms. These outstanding features that may be on the way to adaptability are partly lost in the existing learning algorithms, such as the PC algorithm (Spirte *et al.* 2000) and the MMBN algorithm (Tsamardinos *et al.* 2003). Recently, the block learning algorithms (Zeng & Poh 2004) just demonstrates its good performance on learning large Bayesian networks from sparse data. Its generalization would hit the target of adaptability.

Generalize the Block Learning Algorithm

Specified on learning Bayesian network structures, the block learning algorithm adopts divide-and-conquer strategy to decompose a learning problem into several sub-learning tasks. Its generalization includes some major procedures: (1) Generate Maximum Spanning Tree (GMST); (2) Identify Blocks (IB); (3) Identify Markov Blanket of Overlaps (IMB); (4) Learn Overlaps (LO); (5) Learn Blocks (LB); (6) Combine Learned Blocks (CB).

Generate Maximum Spanning Tree (GMST). In the GMST procedure shown in Figure 1, an MST is built after computing the mutual information between every pair of variables $\chi = \{X_1, \cdots, X_n\}$ from a training data set $D = \{x^1, \cdots, x^N\}$. This procedure provides an initial dependency structure with little computation cost.

Procedure GMST
Input: A data set $D = \{x^1, \cdots, x^N\}$
Output: MST M

1. Load the data set D
2. Build M based on the mutual information

Figure 1: GMST Procedure

Identify Blocks (IB). In the IB procedure shown in Figure 2, a block is a kind of dense structure that includes nodes with strong dependency. The number of the shared nodes $ComNum1$ and $ComNum2$ determines the criteria to merge these rough blocks. The finalized block becomes a basic leaning unit in a learning process.

Identify Markov Blanket of Overlaps (IMB). In the IMB procedure shown in Figure 3, what we are interested in is the Markov blanket of overlaps, not the overlaps that link the related blocks. Here, a simple way is formulated.

Learn Overlaps (LO). In the LO procedure shown in Figure 4, any learning algorithm ALG1 could be utilized to learn the Markov blanket of the overlaps. However, only the V-structures associated with the overlaps remain in the next procedure for these robust structures is possible to improve the learning reliability as well as spend up learning efficiency.

Procedure IB
Input: A graph M
Output: Blocks B_i

1. Initialize an individual block B_i ($i = 1 \cdots n$) as one block center S_i with its family $Fam(S_i)$ in M
2. Merge block B_i and B_j ($i \neq j$) that have the same cardinality of connectivity and share the number of nodes larger than $ComNum1$
3. Search leaf nodes connected to those nodes in $Fam(S_i)$ and enclose them into block B_i
4. Merge block B_i and B_j ($i \neq j$) that share the number of nodes larger than $ComNum2$
5. Finalize Blocks B_i ($i = 1 \cdots r$)

Figure 2: IB Procedure

Procedure IMB
Input: Blocks B_i ($i = 1 \cdots r$)
Output: Overlaps O_{ij}, Markov Blankets of Overlaps MB_{ij}

1. Identify O_{ij} between block B_i and B_j ($i \neq j$)
2. Search all nodes within two lengths away from nodes in O_{ij} and pull them into MB_{ij}

Figure 3: IMB Procedure

Procedure LO
Input: A data set $D = \{x^1, \cdots, x^N\}$ and MB_{ij}
Output: V-Structure of O_{ij} : VS(O_{ij})

1. Load the data set D and learn MB_{ij} using ALG1
2. Identify the V-Structure of O_{ij} from the learned MB_{ij}

Figure 4: LO Procedure

Learn Blocks (LB). The procedure LB shown in Figure 5 is the core component in the block learning algorithm. In this procedure, the learning unit is not the whole network but the block obtained in the procedure IB so that the learning efficiency is increased absolutely. At the same time, the learning procedure LB could be configured with any learning algorithm ALG2. Furthermore, the benefit of robust structure in the overlaps is utilized while possible local errors are restricted in the block. These strategies are very helpful in the general learning task.

Procedure LB
Input: A data set $D = \{x^1, \cdots, x^N\}$, B_i and VS(O_{ij})
Output: Learned B_i ($i = 1 \cdots r$)

1. Load the data set D and learn B_i using the ALG2 with constraints VS(O_{ij})
2. Produce the learned B_i ($i = 1 \cdots r$)

Figure 5: IB Procedure

Combine Learned Blocks (CB). The final procedure CB is shown in Figure 6. The challenging work in this procedure is to ensure the final combined structure is a real Bayesian network of acyclic directed graph. Some strict combination methods could be found in (Jiang et al. 2005). Here, we borrow the fact that a partial order λ_i equals to λ_j (λ_i , λ_j : a partial order for the variables in O_{ij} identified in the blocks B_i and B_j individually) ensures no directed cycles in B_{ij} composed of B_i and B_j ($i \neq j$). On another aspect, in the case that the reversal orders are obtained in λ_i and λ_j for the variables in O_{ij} , we have to force them to follow one uniform order in the block B_i that has a smaller size m_i , assuming $m_i < m_j$. The reason lies in the consideration of the confounding information in the statistical test: for the same cases or instances, the fewer the variables, the more reliably the dependencies among variables are tested. These strategies support the validity and effectivity of combination methods in this procedure.

It seems that the above generalization parameterizes the block learning algorithm in our recent work (Zeng & Poh 2004), like the introduction of *ComNum*1 and *ComNum*2 in the IB and IMB procedures, ALG1 and ALG2 in the LO and LB procedures. Similarly, the complexity for this generalization algorithm could be analyzed as that in (Zeng & Poh 2004).

Undoubtedly, the generalization of the block learning algorithm still keeps scalability to learn large Bayesian networks for it learns each block individually instead of learning the whole network simultaneously. At the same time, it can also learn blocks in which designers are interested. Hence, it is also an incremental learning algorithm. In the learning procedures, any type of learning algorithm, like the PC algorithm (Spirtes *et al.* 1993) and the GS algorithm (Margaritis & Thrun 1999), can be used to parameterize ALG1 or ALG2. In other words, the block learning algorithm could be considered as a kind of learning strategy rather than a type of learning algorithm.

Procedure CB
Input: Learned B_i and VS(O_{ij}) ($i, j = 1 \cdots r$ and $i \neq j$)
Output: The whole Bayesian Networks B

1. Identify λ_i , λ_j for the nodes in O_{ij} between block B_i and B_j
2. For the two blocks B_i and B_j ($i \neq j$)
   if $\lambda_i = \lambda_j$, combine B_i and B_j
   else if $m_i < m_j$, combine B_i and B_j
   following λ_i .
3. Produce B

Figure 6: CB Procedure

Integration with Grid

Learning Bayesian network, especially learning a large one, is a time-consuming task that always frustrates practitioners in the relevant domain, like in the computation biology field. Some work, like the parallel or distributed learning Bayesian network in (Xiang & Chu 1999, Lam & Segre 2002), has facilitated the learning task a bit. With the emergence of advanced computing techniques, like grid computing, it is promising that the learning algorithm could share the existing computational advantages.

It is evident that a learning task armed with this generalization for the block learning algorithms is easily to be gridified shown in Figure 7. In Figure 7, an oval with a solid line indicates a block obtained in the learning process; while an oval with a dotted line indicates an overlap linking the adjacent blocks in which the linkage nodes are denoted with solid circles. Obviously, the procedures for learning different overlaps or blocks can be done with distributed resources on the grid simultaneously as shown in Figure 7. In Figure 7, the machine-like symbols not only indicate the PC machines or workstations, but also indicate all the available resources on the grid. This framework can reduce a large amount of execution time if learning tasks are assigned to computing nodes and scheduled properly.

Accordingly, the block learning algorithm could exerts most of its advantages when it is integrated with grid computing. However, much challenging work still remains to be addressed: (1) To assign and schedule the learning tasks to computing nodes properly, the computation cost and the communication cost of learning tasks must be known in advance. It requires a detailed analysis on the complexity of learning algorithms in grid experiments. (2) The criteria to obtain blocks have to be reconsidered according to computational resources on the grid, like the sections of $ComNum1$ and $ComNum2$. It is not a fact that the larger the number of blocks, the more efficient the block learning algorithm. Hence, a parameterized block learning algorithm has to be tested.

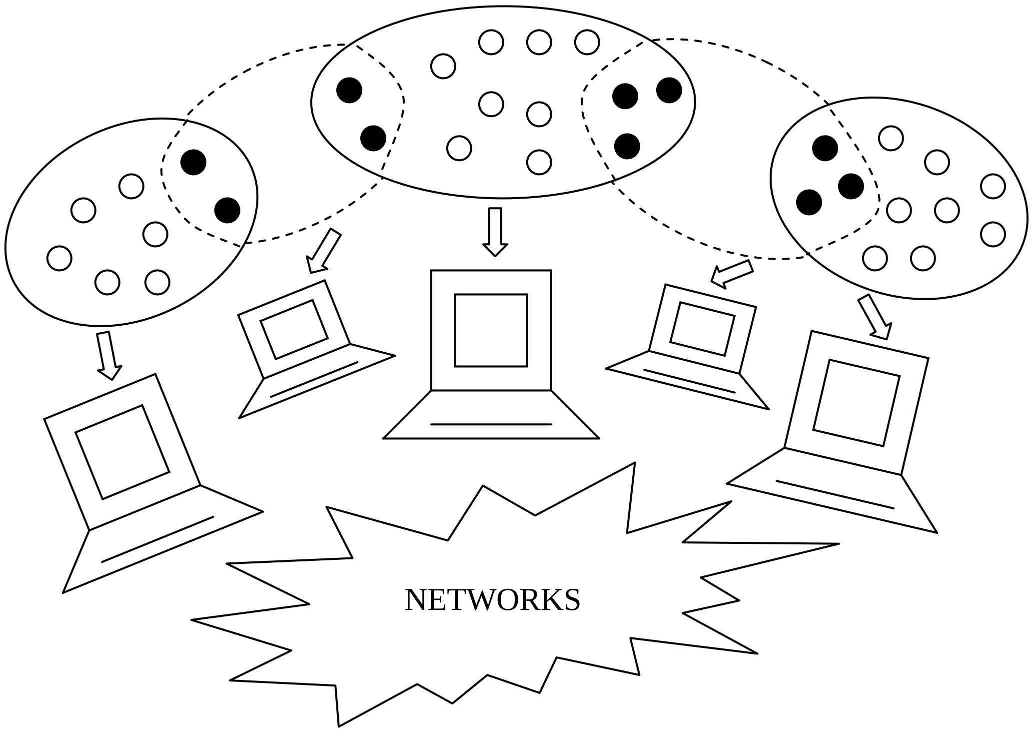

Figure 7: Block Learning Algorithm on the Grid

Grid-Enabled Bayesian Inferential Methods

The major contribution of Bayesian network to decision support systems rests heavily on its accurate prediction through some effective inference methods. Tracing the development of inference algorithms for Bayesian networks, we find that these algorithms unconsciously embrace the thinking of adaptability. For instance, junction tree algorithms (Jensen *et al.* 1990) based on the message passing are a natural fit for parallel or distributed systems, like the work in (Madsen & Jensen 1999) and (D'iez & Mira 1994). In a fine-grain view, cliques in a junction tree are independent of each other and have their own local probabilities. After the message passing through these cliques, the junction tree algorithm produces precise propagation results. In a coarse-grain view, a junction tree consists of several branches composing of cliques that have strong dependency. Hence, each branch can be considered as one local probabilistic unit. Communications occur only in the joint cliques, which results in coherent inferences.

In any case, junction tree algorithms can be gridified naturally by treating cliques or branches as computing nodes with accessible resources on the grid. For example, in Figure 7, an oval with a solid line could represent a clique obtained in a junction tree; while an oval with a dotted line may represent joint cliques. In this way, the propagation in local cliques occur sequentially or simultaneously concerning their partial orders generated in the junction tree so that it could make full use of the available computational recourses on the grid. This integration will make inference algorithms more practical, especially reasoning in large Bayesian networks, because enormous computational resources are fully used to steer a time-consuming propagation. However, some problems arise when inference is performed in a grid computing environment, such as communication or computing nodes failure leading to non-global reasoning results. Hence, inference algorithms need to be improved and configured on the grid. This work requires collaboration between algorithms designers and software engineers.

Recently, some progress has been achieved on new inference algorithms, called robust message passing algorithm (Paskin & Guestrin 2004). This advanced algorithm seems to have overcome a communication failure in a propagation process and provided very strong theoretical guarantees. Till now, most of this work has emphasized the algorithm formulation and theory discussion. To achieve practical adaptive systems, challenging work still exists. First, the algorithm has to be generalized and its performance should be verified in the follow-up tests. Second, some strategies must be proposed with the consideration on the grid reliability. For instance, how to handle an underway propagation when computing nodes involved in the computation are lost? What data related to the propagation should be kept? How to activate a new round of inference? Third, the algorithms should be able to adapt to grid environments, where the communication rate among computing nodes may be not high. In this grid environment, the communication cost is high especially when the large amount data need to be transferred. The high communication cost may be the bottleneck for the performance. To improve the performance, the grid-enabled inferential algorithms should try to achieve a tradeoff between the size of cliques and the number of the cliques.

Conclusion

Grid-enabled Bayesian networks that are realized with grid-enabled learning methods and grid-enabled inferential methods are the integration of Bayesian methods with grid computing techniques. Grid-enabled learning methods are proposed based on the generalization of the block learning algorithm in our current work (Zeng & Poh 2005). This paper emphasizes the generalization of the block learning algorithm. We show this generalization still remains its scalability and further improves its adaptivity. Therefore, its integration with grid computing techniques is direct and effective. However, much challenging work requires more effort. As for the grid-enabled inferential methods, we uncover the essence of the junction tree inference algorithm and discuss its natural and possible integration with grid computing techniques. The latest work on this topic is also valued in this paper.

In a summary, to solve a complicated intelligent task, much work on both Bayesian methods and computing technologies has followed some common advanced ideologies that are discovered from our insightful analysis. Their marriage will be able to produce a powerful capability to cope with some of the practical problems in the changing world. Utilizing all of available computation resource on the grid, grid-enabled Bayesian network will be able to speed up the computation involved in the reasoning and learning process. Hence, it will allow users to be fully prepared for any task in the changing world. Moreover, it will facilitate the building of intelligent and adaptive Bayesian tools. However, lots of arduous and creative works are required to address some challenges in their integration on the grid.

Acknowledgments

The authors would like to thank Marek J. Druzdzel (at the University of Pittsburgh, U.S.A.) for his discussions on the block learning algorithms, and Tze-Yun Leong and members of Biomedical Decision Engineering Research Group at the National University of Singapore for their helpful comments on the initial manuscript.

References

D'iez, F. J. and Mira, J. 1994. Distributed inference in Bayesian networks. *Cybernetics and Systems* 25:39-6.

Foster, I., Kesselman, C., Nick, J. and Tuecke, S. 2002. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, *Open Grid Service Infrastructure WG*, Global Grid Forum.

Jiang, C.A., Poh, K.L. and Leong, T.Y. 2005. Integration of Probabilistic Graphic Models for Decision Support, In the Proceedings of the 2005 AAAI Spring Symposium on Challenges to Decision Support in a Changing World. March 21-23, 2005, Stanford University, to appear.

Jensen, F. V., Lauritzen, K. G. and Olesen, K. G. 1990. Bayesian Updating in Causal Probabilistic Networks by Local Computations. *Computational* Statistical Quarter 4: 269-282.

Korb, K. B. and Nicholson, A. E. 2004. *Bayesian Artificial Intelligence*, Chapman & Hall/CRC.

Lam, W. and Segre, A. M. 2002. A distributed learning algorithm for Bayesian inference networks. *IEEE Transactions on Knowledge and Data Engineering* 12(1): 93-105.

Madsen, A. L. and Jensen, F.V. 1999. Parallelization of Inference in Bayesian Networks, Research Report R-99-5002, Department of Computer Science, Aalborg University.

Margaritis, D. and Thrun, S. 1999. Bayesian Networks Induction via Local Nerghborhoods, In *Advances in Neural Information Processing Systems12 (NIPS).*

Paskin, M. A. and Guestrin, C. E. 2004. Robust Probabilistic Inference in Distributed Systems, In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, 436-445.

Spirtes, P., Glymour. G. and Scheines, R. 2000. *Causation, Prediction and Search*. Cambridge, Mass, MIT Press.

Tsamardinos, I., Aliferis, C. F., Statnikov, A. and Brown, L. E. 2003. Scaling-Up Bayesian Network Learning to Thousands of Variables Using Local Learning Technique, DSL TR-03-02, Vanderbilt University, Nashville, TN, USA.

Xiang, Y. and Chu, T., 1999. Parallel Learning of Belief Networks in Large and Difficult Domains. *Data Mining and Knowledge Discovery* 3: 315-339.

Zeng, Y. F. and Poh, K. L. 2004. Block Learning Bayesian Network Structures from data, In Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS'04), IEEE Computer Society.