

Loopy Belief Propagation Application for Advanced Decision Making Models

Wiboonsak Watthayu

Department of Mathematics, Applied Computer Science Division,
Faculty of Science, King Mongkut's University of Technology
Bangkok, Thailand
e-mail: iwibhayu@kmutt.ac.th

Abstract—Loopy Belief propagation (LBP) is a technique for distributed inference in performing approximate inference on arbitrary graphical models. In this paper, we propose an extended influence diagram with loopy belief propagation technique to solve the advanced decision making problems, especially the multi-criteria decision making (MCDM) in term of uncertain criteria. When a number of nodes is increasing in the network, solving for exact solutions and making optimal decision becomes complicated and time consuming. The belief propagation is an efficient way to solve inference problems by propagating local messages around neighborhoods. In this paper, we will focus on the principle of loopy belief propagation, and its applications in advanced decision making problems. Loopy belief propagation is another choice for solving the advance decision making problem in order to get approximate solution and is selected to present in this paper. A solution is approximated which a high probability action under the preference criteria and high utility. In addition, the comparison between the loopy belief propagation method including other approximated method and the exact methods based on the prediction and the running time is shown in this research.

Keywords- Loopy Belief propagation; Influence Diagram; Multi-criteria Decision Making

I. INTRODUCTION

The Belief propagation algorithm is well studied in the literature and is currently used in order to analyze systems which are modeled by decision support system. The belief propagation algorithm allows us to get the marginal of all variables, with a computational time only in the linear in the number of variables. This algorithm can be used in a vast range of decision making model areas, especially, in Multi Criteria Decision-Making Method (MCDM). This method is about to make preference decisions over the available alternatives based on multiple and conflicting criteria [1]. In previous research, the model of decision-making is complicated because there are a lot of choices to choose from and the people have many factors to be considered while making the decision. Therefore, the advance techniques for solving the decision need to be investigate in order to get the optimal solution or the to make the right decision for decision makers.

Bayesian network and Influence diagram (ID) are developed in the field of artificial intelligence (AI) for representing and reasoning under uncertainty. In the model,

decision criteria and factors are considered as random variables and represented as node in the network. The network structure associated with conditional probability tables and each node which provide a representation of the joint probability distribution of all variables. There are some algorithms that have been developed for probabilistic inference like probabilistic inference, Bayesian network. This paper focuses on the method for solving advanced decision making problems and presents our current research for developing an approximate algorithm for solving advanced decision making problems.

Nowadays, it appears many algorithms in both exact and approximate, for solving advanced decision making problems. Normally, researchers like to use approximate methods such as Monte Carlo (MCMC) simulation, sampling method, variational method, and so on for solving the decision problems. However, those methods are very complicated and hard to get both exact and approximated solutions. Sallans [12] found that the sampling method and MCMC method is good solution when the number of nodes is increase or in the large the network. However, MCMC and sampling method have a convergence problem and they can apply for a limited in some networks [14].

In this research loopy belief propagation is proposed for finding the solution to evaluate the advanced decision making problems. It has been used for approximate inference in a wide variety of many model especially in Bayesian network and influence diagram model [14]. The loopy propagation method is compared to others methods in prediction and performance of running time. Therefore, the likelihood weight method, junction tree method and Markov Chain Monte Carlo (MCMC) method are selected for experiment in this paper.

II. BACKGROUND

The advance decision making model becomes more complicated when we cannot find the solutions. To solve these issues, the approximated algorithm has been developed based on loopy belief propagation. There have some approximate algorithms for solving the problems such as sampling, and variational methods. However, these algorithms are still have the problem in term of complexity time when the problem becomes large and complicated.

Advanced decision making refers as MCDM. This model is about to making decisions among multiple criteria. The model of MCDM problem is formed into hierarchy

which consists four elements: the *goal*, the *objectives*, the *criteria*, and the *alternatives*. Let $A = \{A_1, \dots, A_m\}$ be a set of decision alternatives and $C = \{c_1, \dots, c_n\}$ be a set of criteria. The elements form into a decision matrix D is an $m \times n$ matrix, where element d_{ij} indicates the performance of alternative a_i when it is evaluated in terms of decision criteria c_j . The decision maker can determined the weights of relative importance of the decision criteria, $W = \{w_1, \dots, w_n\}$ [10] with the assumption. The total score for each alternative is calculated by the following formula:

$$A_i = \sum_j w_j d_{ij}$$

Once all scores are calculated, the highest score is selected among the overall scores. In the past, many MCDM methods have been developed [3] and the multiple criteria and multiple objectives have been considered to solve the problems. However, there still have some issues that cannot be solved since the current problems are more and more complicated. In the next section, the proposed method to solve the complicated problem is discussed.

III. METHODOLOGY

A. Influence Diagram

An influence diagram (ID) is the way to graphically depicting the relationship between several elements that can affect the results of a decision. ID gives decision makers an understanding of the results of actions they may choose. But, it is only as accurate as the insights the decision makers bring to the situation. The algorithms of ID propagate beliefs and update the beliefs in each node in the network. The way to propagate can be exact algorithms or approximate algorithms. A more detailed about ID can be found in Castillo [4].

1) *Exact Algorithms*. Junction tree algorithm is one of the most widely used for exact algorithm. It was developed by Jensen, Lauritzen [5, 6, & 8]. The concept of junction tree is based on a deep analysis of the connections between graph theory and probability theory. It uses an auxiliary data structure called junction tree. It can be constructed along the process of moralization and triangulation [6 & 8]. Construct the junction tree, starting with cliques as the nodes where each link between two cliques is labeled by a separator set. The junction tree states that any path through the junction tree from node X to node Y must contain nodes labeled by cliques containing all vertices $X \cap Y$.

2) *Approximate Algorithms*. This research is about approximate inference in general discrete probabilistic models. Monte Carlo algorithm is the popular approximate algorithm for inference and can handle the complicated problems. This method generates random selected samples of the network. This method has separated in two groups as importance sampling algorithms and Markov Chain Monte Carlo (MCMC) methods.

Jian Cheng[16] proposed the adaptive importance sampling for BNs (AIS-BN). This method is more efficient

technique. It can reduce the sampling variance. The concept of AIS-BN algorithm is applying different weights for samples and it generates the weight in different learning stages. In Cheng [16], AIS-BN works well for large networks. However, this algorithm is too complicated and it is not good in running time.

Another technique is MCMC which belongs to group of stochastic sampling algorithms. This method can approximate the solution by generating independent different sampling. Therefore, it can estimate of value when the approximated solutions compare to the other exact solutions but it is not good for the response time to response the solutions.

The loopy belief propagation can perform well in approximate solutions. Pearl's polytree propagation algorithm is applied to use in this algorithm as can be seen in the following section which is the extended influence diagram with loopy belief propagation.

B. Extended ID with Loopy Belief Propagation

In loopy belief propagation, the belief $BEL(x)$ can be defined a node x in a graph as follows:

$$BEL(x) = P(x|e) = P(x|e^+, e^-)$$

$$\begin{aligned} &= P(e^-|x, e^+) * P(x|e^+) / P(e^-|e^+) \\ &= \alpha P(e^-|x) P(x|e^+) = \alpha \lambda(x) \pi(x) \end{aligned}$$

For polytrees or network containing no loops, given a node X having parents U_1, \dots, U_m and children Y_1, \dots, Y_n , E is the evidence above X as e^+ and the evidence below X as e^- . The π and λ are the messages. We need to compute the belief of X was computed in the value of true or false ($BEL(X = \text{true})/BEL(X = \text{false})$). The belief of X as shown in (1) can be given its parent and child values.

$$BEL(x) = P(X|e) = \alpha \lambda(X) \pi(X) \quad (1)$$

Where α is a normalizing value, $\lambda(X) = P(e^-/x)$, and $\pi(X) = P(x/e^+)$.

$$\lambda(X) = \prod_{i=1}^j \lambda_{Y_i} \quad (2)$$

$$\pi(X) = \sum P(x/\bar{u}) \pi(\bar{u}) \quad (3)$$

where U is a vector of values for U_1, \dots, U_m . The X have to send π and λ messages to its' children and parents when $BEL(X)$ is updated. However, Y_i 's belief has to be updated based on evidence from Y_i and the same for U_i . This is done by calculating the messages to send to Y_i and U_i as follows:

$$\pi(Y_i) = \frac{BEL(X)}{\lambda_{Y_i}} \quad (4)$$

and

$$\lambda(U_i) = \beta \sum_X \lambda(X) \sum_{u_k: k \neq i} P(X/\bar{u}) \prod_{k \neq i} \pi(u_k) \quad (5)$$

Where β is a normalizing factor and \bar{u} is the vector of values for U_1, \dots, U_m . Therefore, each node tells its neighbors what it believes, based on prior probability and the belief of its neighbors. For more details of Pearl's algorithm, see [2].

There are 4 steps for the algorithm:

1. Initial Conditions

If X has no parents, initial with the prior:

$$\pi(x) = \Pr[X = x]$$

If X is an observed node with value e ,

$$\lambda(x) = 1 \text{ when } x = e,$$

$$\lambda(x) = 0 \text{ otherwise}$$

If X is not observed and has no children, $\lambda^{(0)}(x) = 1$

2. Beliefs were updated

3. Propagation of belief change

For a node X with parents $U = \{U_1, \dots, U_n\}$ and

children $Y = \{Y_1, \dots, Y_m\}$:

Incomming message to node X at time t :

$$\lambda^{(t)}(x) = \lambda_x(x) \prod_j \lambda_{Y_j}^{(t)}$$

$$\pi^{(t)}(x) = \sum_u \Pr[X = x | U = u] \lambda \prod_k \pi_{X_j}^{(t)}(u_k)$$

$$\lambda^{(t)}(u_i) = \alpha \sum_x \lambda^{(t)}(x) \Pr[X = x | U = u] \lambda \prod_k \pi_{X_j}^{(t)}(u_k)$$

$$\pi^{(t+1)}(x) = \alpha \pi^{(t)}(x) \lambda_x(x) \prod_j \lambda_{Y_j}^{(t)}(x)$$

4. If beliefs changed, loop on step 2

Firstly, a single loop was considered at a network in loopy belief propagation. The belief will propagate around the loop. The graph with single loop can be found in [5].

IV. EXPERIMENTS AND RESULTS

This section shows the experimental results: a comparison of exact algorithm and approximated algorithms in term of relative error of prediction and running time. In the experiment and results, we randomly created the number of nodes start from five nodes up to fifty nodes. More than fifty nodes are also investigated but not include in the experiments. The experiments and results can be shown in the graph. Figure 1 shows the performance of prediction in variety number of nodes from 5, 10, ..., 50 nodes where we collected in term of the relative error of the prediction in each experiment with the same sample data. We run this experiment with the same problem for four algorithms (loopy belief propagation, MCMC, AIS-BN, and exact algorithm). We found that loopy belief propagation shows the solution that is closed to the exact solution. The figure 2 shows the running time for each algorithm when queries of the solutions in the same decision making problems with four algorithms.

The experiments were run by using exact algorithms (junction tree and likelihood) and approximate algorithm (Loopy Propagation, MCMC, AIS-BN). Some decision making problems were used for the data in the experiments. We created the IDs in three categories where the small-sized (10 nodes), the medium-sized (20 nodes) and large-sized (50 nodes and more). The selected exact algorithms (likelihood weighting and junction tree) and the approximate algorithms (Loopy propagation, MCMC, AIS-BN) were tested by using three groups for prediction and running time. The results can be shown in figure 1 and figure 2 respectively.

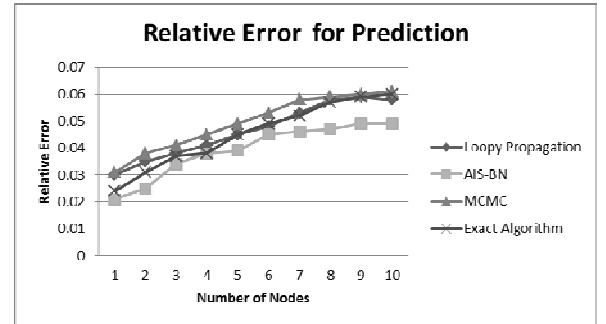


Figure 1 The Relative errors for prediction

In figure 1, the number of nodes represents in the 1 means 10 nodes when run in the experiments. Therefore, we use 100 nodes for testing in our experiments for the good results. In the relative error for prediction, the loopy propagation shows that the relative error is much closed to the exact algorithm in figure 1 when compare with other approximated algorithms. In figure 2, the exact algorithm shows the best running time for the small-sized network. However, this does not work with large-sized especially when the number of nodes is 35 nodes and up. Therefore, the approximate algorithm is good for the large-sized network in running time test.

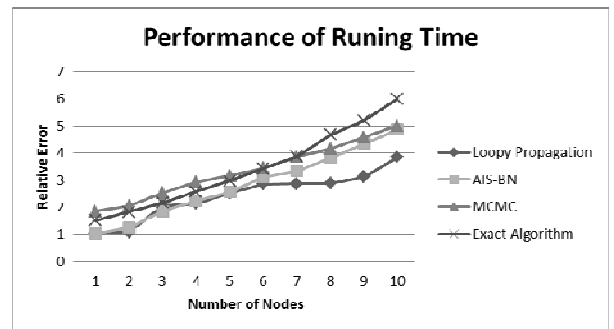


Figure 2 Performance of Running Time

In our experiments, we conducted the set of tests by randomly generated the different thirty influence diagrams. The assumptions were each of influence diagrams consist only one utility node, many decision nodes, and chance nodes. However, conducting experiments with large

network is challenging. On 30 nodes and up random graph, LBP shows a good performance in both prediction and running times.

V. CONCLUSION

The methodology of loopy belief propagation for approximated solution is presented in the paper. The exact methods show a good running time with small-sized network with the exact solutions. However, the exact methods are not suitable with large number of nodes e.g. more than 30 nodes. In this paper, the approximate methods are good performance in running time with a large-sized network but the approximated solutions can only get here. The loopy belief propagation shows a very distinct technique when the networks become large. Therefore, the approximate methods can be a good to solve the advanced decision making problem because the complicated problems are always formed in the large-sized. Finally, loopy belief propagation is not yet fully understood from a theoretical point of view. On very large or complicated problem, we try to work it out in our ongoing research. We hope this method would scale much better.

ACKNOWLEDGMENT

This research was supported by the National Research Council of Thailand (NRCT) grant and department of Mathematics, Faculty of Science, King Mongkut's

University of Technology Thonburi (KMUTT), Bangkok, Thailand.

REFERENCES

- [1] J. Pearl, J., (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, ISBN 1-55860-479-0, 1988.
- [2] Stewart T. J., "A critical survey on the status of multiple criteria decision making theory and practice," *OMEGA International*, 1992.
- [3] Castillo, E., Gutierrez, J. M., and Hadi, A. S., *Expert Systems and Probabilistic Network Models*: Springer, 1997.
- [4] Shafer, G., *Probabilistic Expert Systems*, CBMS-NFS regional Conference Series in Applied Mathematics; SIAM, 1996.
- [5] Vincke, P., *Multi-criteria Decision-Aid*, Wiley, Chichester, 1992.
- [6] Madson, A. L. and Jensen, F. V., "Lazy propagation in junction trees", in proceedings of the 14th Conference on uncertainty in Artificial Intelligence, 1998.
- [7] Murphy, K. P., "Software Packages for Graphical Models: Bayesian Networks", <http://www.cs.berkeley.edu/~murphyk/Bayes/bnsoft.html>, 31 March 2007.
- [8] Xiang, Y., *Probabilistic Reasoning in Multiagent Systems: A Graphical Model Approach*, Cambridge University Press, 2002.
- [9] Sallans, B., "Variational Action Selection for Influence Diagrams," *OEFAI-TR-2003-29*, 2003.
- [10] Weiss, Y. and Freeman, W. "Correctness of belief propagation in Gaussian graphical models of arbitrary topology." *Neural Computation*, 2006.