TESTING AND IMPROVING LOCAL ADAPTIVE IMPORTANCE SAMPLING IN LJF LOCAL-JT IN MULTIPLY SECTIONED BAYESIAN NETWORKS

Dan Wu[1] and Sonia Bhatti[2]

[1]School of Computer Science University of Windsor, Windsor, Ontario Canada
danwu@uwindsor.ca
[2]School of Computer Science University of Windsor, Windsor, Ontario Canada
bhattif@uwindsor.ca

ABSTRACT

Multiply Sectioned Bayesian Network (MSBN) provides a model for probabilistic reasoning in multi-agent systems. The exact inference is costly and difficult to be applied in the context of MSBNs as the size of problem domain becomes larger and complex. So the approximate techniques are used as an alternative in such cases. Recently, for reasoning in MSBNs, LJF-based Local Adaptive Importance Sampler (LLAIS) has been developed for approximate reasoning in MSBNs. However, the prototype of LLAIS is tested only on Alarm Network (37 nodes). But further testing on larger networks has not been reported yet, so the scalability and reliability of algorithm remains questionable. Hence, we tested LLAIS on three large networks (treated as local JTs) namely Hailfinder (56 nodes), Win95pts (76 nodes) and PathFinder(109 nodes). From the experiments done, it is seen that LLAIS without parameters tuned shows good convergence for Hailfinder and Win95pts but not for Pathfinder network. Further when these parameters are tuned the algorithm shows considerable improvement in its accuracy and convergence for all the three networks tested.

KEYWORDS

MSBN, LJF, Adaptive Importance sampling, Tunable parameters

1. INTRODUCTION

Multiply Sectioned Bayesian Networks (MSBN) is the model grounded on the idea of cooperative multi-agent probabilistic reasoning, is an extension of the traditional Bayesian Network model and it provide us with solution to the probabilistic Reasoning under cooperative agents. The Multiple agents [1] collectively and cooperatively reason about their respective problem domain on the basis of their local knowledge, local observation and limited inter-agent communication. Typically the inference in MSBN is generally carried out in some secondary structure known as *linked Junction tree forest* (LJF). The LJF provides a coherent framework for exact inference with MSBN [2], LJF constitutes local Junction trees (JT) and linkage trees for making connections between the neighbouring agents to communicate among themselves. Agents communicate through the messages passed over the LJF linkage trees and belief updates in each LJF local junction tree (JT) are performed upon the arrival of a new inter-agent message.

Jan Zizka et al. (Eds) : CCSEIT, AIAP, DMDB, MoWiN, CoSIT, CRIS, SIGL, ICBB, CNSA-2016
pp. 67–78, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60606

However the computational cost of exact inference makes it impractical for larger and complex domains. So the approximate inference algorithms are being used to estimate the posterior beliefs. Hence, it is very important to study the practicability and convergence properties of sampling algorithms on large Bayesian networks.

To date there are many stochastic sampling algorithms proposed for Bayesian Networks and are widely used in BN approximation but this area is quite problematic, since many attempts have been made in developing MSBN approximation algorithms but all of these forgo the LJF structure and sample MSBN directly in global context. Also it has been shown that such type of approximation requires more inter-agent message passing and also leaks the privacy of local subnet [3]. So, sampling MSBN in global context is not good idea as it analyses only small part of entire multi-agent domain space. So in order to examine local approximation and to maintain LJF framework, the sampling process is to be done at each agent's subnet. The LJF-based Local adaptive Importance Sampler (LLAIS) [3] is an example of extension of BN Importance sampling techniques to JT's. An important aspect of this algorithm is that it facilitates inter-agent message calculation along with the approximation of the posterior probabilities.

So far the application of LLAIS is done on smaller network consisting of 37 nodes which is treated as local JT in LJF. LLAIS produced good estimates of local posterior beliefs for this smaller network but its further testing on larger sizes of local JTs is not reported yet. We tested LLAIS for its scalability and reliability on the three larger networks treating them as local JTs in LJF. It is important to test the algorithm since the size of local JT can vary and can go beyond 37 nodes network, on which preliminary testing has been done. Our testing demonstrated that without tuning of parameters, LLAIS is quite scalable for Hailfinder (56 nodes) and Win95pts (76 nodes) but once it is applied to Pathfinder (109 nodes) network its performance deteriorates. Further, when these parameters are tuned properly it resulted in significant improvement in the performance of algorithm, now it requires less number of samples and less updates than required by the original algorithm to give better results.

2. BACKGROUND

2.1 Multiply Sectioned Bayesian Networks (MSBNs)

In this paper, we assume that the reader is familiar with Bayesian networks (BNs) and basic probability theory [4]. The Multiply Sectioned Bayesian Networks (MSBNs) [2] extend the traditional BN model from a single agent oriented paradigm to the distributed multi-agent paradigm and provides a framework to apply probabilistic inference in distributed multi-agent systems. Under MSBNs, a large domain can be modelled modularly and the inference task can be performed in coherent and distributed fashion.

The **MSBN model** is based on the following five assumptions:

1. Agent's belief is represented as probability.

2. Agents communicate their beliefs based on a small set of shared variables.

3. A simpler agent organization is preferred.

4. A DAG is used to structure each agent's knowledge.

5. An agent's local JPD admits the agent's belief of its local variables and the shared variables with other agents.

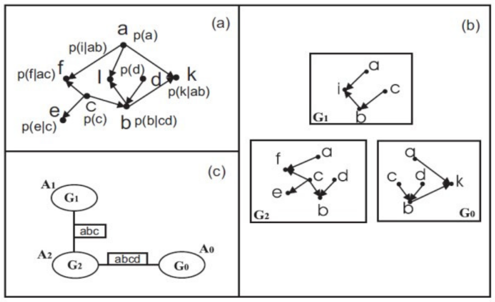

Figure 1: (a) A BN (b) A small MSBN with three subnets (c) the corresponding MSBN hypertree.

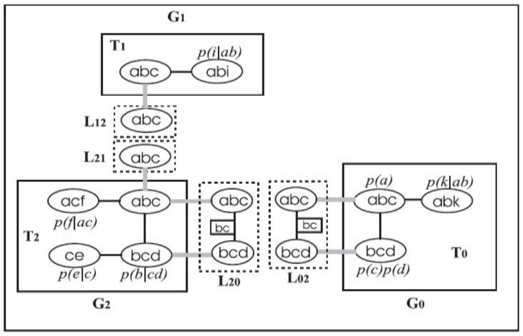

Figure 2. An MSBN LJF shown with initial potentials assigned to all the three subnets.

MSBN consist of set of BN subnets where each subnet represents the partial view of a larger problem domain. The union of all subnet DAGs must also be DAG, denoted by G. These subnets are organised into a tree structure called a *hypertree* [2] denoted by ψ. Each hypertree node, known as *hypernode*, corresponds to a subnet; each hypertree link, known as *hyperlink*, corresponds to a *d-sepset*, which is set of shared variables between the adjacent subnets. A hypertree ψ is purposely structured so that (1) for any variable x contained in more than one subnet with its parents $\pi(x)$ in G, there must exist a subnet containing $\pi(x)$; (2) shared variables between two subnets N_i and N_j are contained in each subnet on the path between N_i and N_j in ψ. A hyperlink renders two sides of the network conditionally independent similar to the separator in a junction tree (JT).

Fig. 1 (a) shows BN which is sectioned into MSBN with three subnets in Fig. 1(b) and Fig. 1(c) shows the corresponding *hypertree* structure. A derived secondary structure called *linked junction tree forest*(LJF) is used for inference in MSBNs; it is constructed through a process of cooperative and distributed compilation where each *hypernode* in *hypertree* ψ is transformed into local JT, and each hyperlink is transformed into a *linkage tree*, which is a JT constructed from d-sepset. Each cluster of a linkage tree is called a *linkage*, and each separator, a *linkage separator*. The cluster in a local JT that contains a linkage is called a *linkage host*. Fig. 2 shows the LJF constructed from the MSBN in Fig.1 (b) and (c). Local JTs, T_0, T_1 and T_2 are constructed from BN subnets G_0, G_1 and G_2 respectively, are enclosed by boxes with solid edges. The linkage trees; $L_{20}(L_{02})$ and $L_{21}(L_{12})$, are enclosed by boxes with dotted edges. The linkage tree L_{20} contains two linkages $\{a,b,c\}$ and $\{b,c,d\}$ with linkage separator bc (not shown in the figure). The linkage hosts of T_0 for L_{02} are clusters $\{a,b,c\}$ and $\{b,c,d\}$.

3. BASIC IMPORTANCE SAMPLING FOR LJF

Here we assume that readers are aware of basic importance sampling for LJF local JT. The research done so far has highlighted the difficulties in applying stochastic sampling to MSBNs at a global level [5]. Direct local sampling is also not feasible due to the absence of a valid BN structure [3]. However, an LJF local JT can be calibrated with a marginal over all the variables [6] making local sampling possible. Algorithms proposed earlier combine sampling with JT belief propagation but do not support efficient inter-agent message calculations in context of MSBNs.

The [3] introduced a JT-based importance sampler by defining an explicit form of the importance function so that it facilitates the learning of the optimal importance function. The JPD over all the variables in a calibrated local JT can be obtained similar to Bayesian network DAG factorization.

Let $C_1, C_2 \ldots\ldots, C_m$ be the m JT clusters given in the ordering which satisfies the running intersection property. The separator $S_i = \varnothing$ for $i = 1$ and $S_i = C_i \cap (C_1 \cup C_2 \cup \ldots\ldots \cup C_{i-1})$ for $i = 2,3,\ldots, m$. Since $S_i \subset C_i$, the residuals are defined as $R_i = C_i \setminus S_i$. The junction tree running intersection property guarantees that the separator S_i separates the residual R_i from the set $(C_1 \cup C_2 \cup \ldots\ldots \cup C_{i-1}) \setminus S_i$ in JT.

Thus applying the chain rule to partition the residues given by the separators and have JPD expressed as $P(C_1, \ldots\ldots, C_m) = \prod_{i=1}^{m} P(R_i \mid S_i)$. The main idea is to select the root from the JT clusters and then directing all the separators away from the root forming a directed sampling JT. It is analogous to BN since both follow recursive form of factorization.

Once the JPD has been defined for LJF local JT, the importance function P' in basic sampler is defined as:

$$P'(X \setminus E) = \prod_{i=1}^{m} P(R_i \setminus E \mid S_i) \mid_{E=e} \qquad (1)$$

The vertical bar in $P(R_i \setminus E \mid S_i) \mid_{E=e}$ indicates the substitution of **e** for **E** in $P(R_i \setminus E \mid S_i)$. This importance function is factored into set of local components each corresponding to the JT clusters. It means when the calibrated potential is given on each JT cluster C_i we can easily compute for every cluster the value of $P(R_i \mid S_i)$ directly. For the root cluster: $P(R_i \mid S_i) = P(R_i) = P(C_i), i = 0$.

We traverse a sampling JT and sample variables of the residue set in each cluster corresponding to the local conditional distribution. This sampling is similar to the BN sampling except now group of nodes are being sampled and not the individual nodes. Whenever cluster is encountered with the node in the evidence set **E**, it will be assigned value which is given by evidence assignment. A complete sample consist of the assignment to all the non- evidence nodes according to the local JT's prior distribution.

The score for each sample can be computed as:

$$Score_i = \frac{P(S_i, E)}{P'(S_i)} \qquad (2)$$

The score so computed in Equation 2 will be used in LLAIS algorithm for adaptive importance sampling. It is proven that the optimal importance function for BN importance sampling is the posterior distribution $P(X \mid E = e)$ [7]. Applying this result to JTs, we can define the optimal importance function as:

$$\rho(X \setminus E) = \prod_{i=1}^{m} P(R_i \setminus E \mid E = e) \qquad (3)$$

The above Equation 3 takes into account the influence of all the evidences from all clusters in the sample of current cluster.

3.1 LJF-Based Local Adaptive Importance Sampler (LLAIS)

In 2010, LJF local JT importance sampler called LLAIS [3] was designed that follows the principle of adaptive importance sampling for learning factors of importance function. This algorithm was specifically proposed for the approximation of posteriors in case of local JT in LJF providing the framework for calculation of inter-agent messages between the adjacent local JTs.

The sub-optimal importance function used for LJF Local Adaptive Importance Sampling is as follows,

$$\rho(X \setminus E) = \prod_{i=1}^{m} P(R_i \setminus E \mid S_i, E = e) \qquad (4)$$

This importance function is represented in the form of set of local tables. This importance function is learned to approach the optimal sampling distribution.

These local tables are called the ***Clustered Importance Conditional Probability Table*** **(CICPT).** These CICPT tables are created for each local JT cluster consisting of the probabilities indexed by the separator to the precedent cluster (based on the cluster ordering in the sampling tree) and conditioned by the evidence.

For non-root JT clusters, CICPT table are defined in the form of $P(R_i \mid S_i, E)$, and for the JT root cluster, CICPT table are of the form of $P(R_i \mid S_i, E) = P(C_i \mid E)$.

The learning strategy is to learn these CICPT tables on the basis of most recent batch of samples and hence the influence of all evidences is counted through the current sample set. These CICPT tables have the structure similar to the factored importance function and are alike to an ICPT table of Adaptive Importance Sampling of BN in the previous section 4.1 and are updated periodically by the scores of samples generated from the previous tables.

Algorithm for LLAIS

Step 1. Specify the total number of samples M , total updates K and update interval L, Initialize the CICPT tables as in Equation 4.

Step 2. Generate L samples with the scores according to the current CICPT tables. Estimate $P'(R_i \mid S_i, e)$ by normalizing the scores for each residue set given the states of separator set.

Step 3. Update the CICPT tables based on the following learning function [45]:

$$P^{K+1}(R_i \mid S_i, e) = (1-\eta(k))P^{k}(R_i \mid S_i, e) + \eta(k)P'(R_i \mid S_i, e),$$

where $\eta(k)$ is the learning rate.

Step 4. Modify the importance function if necessary, with the heuristic of Є-cutoff. For the next update, go to Step 2.

Step 5. Generate the samples from the learned importance function and calculate scores as in Equation 2.

Step 6. Output the posterior distribution for each node.

In LLAIS the importance function is dynamically tuned from the initial prior distribution and samples obtained from the current importance function are used to refine gradually the sampling distribution. It is well known that thick tails are desirable for importance sampling in BNs. The reason behind it is that the quality of approximation deteriorates in the presence of probabilities due to generation of large number of samples having zero weights [3]. This issue is solved using the heuristic Є-cutoff [7], the small probabilities are replaced with Є if less than a threshold Є, and the change is compensated by subtracting the difference from the largest probability.

4. Improving LLAIS by Tuning The Tuneable Parameters

The tuneable parameters plays vital role in the performance of sampling algorithm. There are many tuneable parameters in LLAIS such as the heuristic value of threshold ∈-cutoff, updating intervals, number of updates, number of samples and learning rate discussed as follows:

1. Threshold ∈-cutoff – it is used for handling very small probabilities in the network. The proper tuning helps the tail of importance function not to decay faster, the optimal value for ∈-cutoff is dependent upon the network and plays key role in getting better precision these experiments with different cut-off values are motivated from [8].

2. Number of updates and updating interval - the number of updates plays an important role in the sense that it denotes how many times the CICPT table has to be updated so that it will result in optimal output and updating interval denotes the number of samples that have to be updated.

3. Number of samples - plays very important role in the stochastic sampling algorithm as the performance of sampling increases with the number of samples. It is always good to have minimum number of samples that can help you reach better output for it will be time and cost efficient

4. Learning Rate - in [7] is defined as the rate at which optimal importance function will be learned as per the formula $\eta(k) = a(\frac{a}{b})^{k/k\max}$, where a = initial learning rate, b = learning rate in the last step, k = number of updates and $k\max$ = total number of updates.

These tuneable parameters are tuned after many experiments in which they were given heuristically different values and then checked for performance. Table 1 shows the comparison of values of various tuneable parameters for original and improved LLAIS.

Table 1: Shows the comparison of values of various tuneable parameters for original LLAIS and improved LLAIS.

Tunable parameters	**Original LLAIS**	**Improved LLAIS**
Number of samples	5000	4500
Number of updates	5	3
Updating interval	2000	2100
Threshold value	Nodes with outcomes <5	Nodes with outcomes < 5
	0.05	0.01
	Nodes with outcomes < 8	Nodes with outcomes < 8
	0.005	0.006
	Else = 0.0005	Else = 0.0005

5. EXPERIMENT RESULTS

We used Kevin Murphy's Bayesian Network toolbox in MATLAB for experimenting with LLAIS. For testing of LLAIS algorithm, the exact importance function is computed, which is considered to be the optimal one and then its performance of sampling is compared with that of approximate importance function in LLAIS. The testing is done on Hailfinder (56 nodes), Win95pts (76 nodes) and Pathfinder (109 nodes), which are treated as local JT in LJF. The approximation accuracy is measured in terms of *Hellinger's distance* which is considered to be perfect in handling zero probabilities which are common in case of BN.

From [8], The Hellinger's distance between two distributions F_1 and F_2 which have the probabilities $P_1(x_{ij})$ and $P_2(x_{ij})$ for state $j(j = 1,2,\mathrm{K}\,\mathrm{K}\,,n_i)$ of node i respectively, such that $X_i \notin E$ is defined as:

$$H(F_1, F_2) = \sqrt{\frac{\sum_{X_i \in N \backslash E} \sum_{j=1}^{n_i} \{\sqrt{P_1(x_{ij})} - \sqrt{P_2(x_{ij})}\}^2}{\sum_{X_i \in N \backslash E} n_i}} \qquad (5)$$

where N is the set of all nodes in the network, E is the set of evidence nodes and n_i is the number of states for node i. $P_1(x_{ij})$ and $P_2(x_{ij})$ are sampled and exact marginal probability of state j of node i.

5.1 Experiment Results for Testing LLAIS

For each of the three networks we generated in total 30 test cases consisting of the three sequences of 10 test cases each. The three sequences include 9, 11 and 13 evidence nodes respectively. For each of the three networks, LLAIS with exact and approximate importance function is evaluated using $M = 5000 samples$. With LLAIS using approximate importance function, the learning function used is $\eta(k) = a(\frac{a}{b})^{k/k\max}$ and set $a = 0.4$ and $b = 0.14$, total updates $K = 5$ and each updating step, $L = 2000$. The exact importance function is optimal hence it does not require updating and learning.

Fig.4 shows the results for all the 30 test cases generated for Hailfinder network. Each test case was run for 10 times and average *Hellinger's distance* was recorded as a function of $P(E)$ to measure the performance of LLAIS as $P(E)$ goes more and more unlikely. It can be seen that LLAIS using approximate importance function performs quite well and shows good scalability for this network.

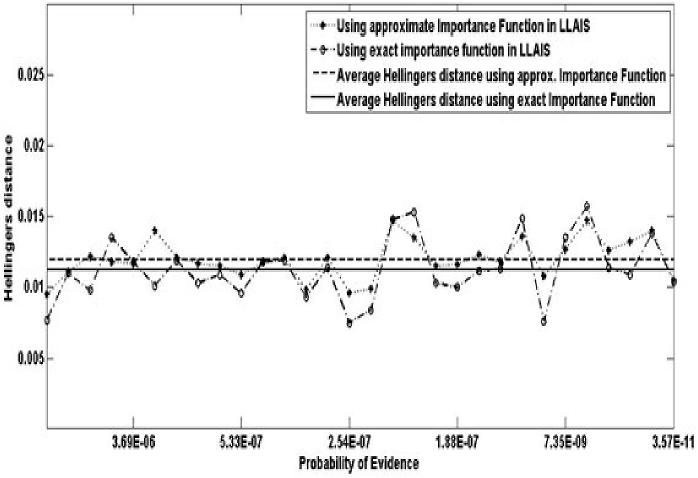

Figure 4: Performance comparison of approximate and exact importance function combining all the 30 test cases generated in terms of Hellinger's distance for Hailfinder network.

Fig. 5 shows the results generated for all the 30 test cases generated from Win95pts network. It can be concluded that for this network too LLAIS using approximate importance function shows good scalability and its performance is quite comparable with that using exact importance function.

Fig. 6 shows the results generated for all the 30 test cases generated from Pathfinder networkIt is seen that for this network LLAIS performed poor, the reason is the presence of extreme probabilities which needs to deal with. Hence LLAIS doesn't prove to be scalable and reliable for this network.

Table 2 below shows the comparison of the statistical results for all the 30 test cases generated using approximate and exact importance function in LLAIS.

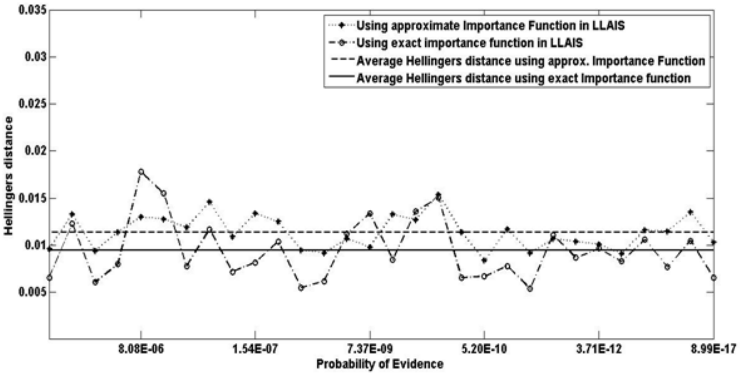

Figure 5: Performance comparison of approximate and exact importance function combining all the 30 test cases generated in terms of Hellinger's distance for Win95pts network.

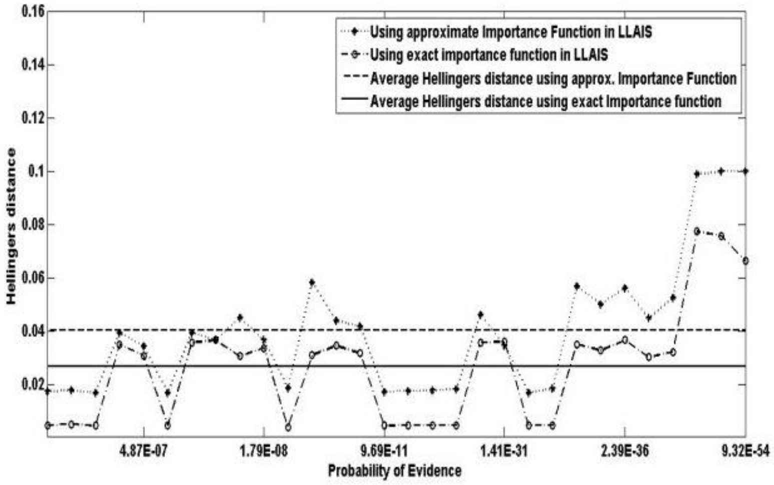

Figure 6: Performance comparison of approximate and exact importance function combining all the 30 test cases generated in terms of Hellinger's distance for Pathfinder network.

Table 2: Comparing the statistical results for all 30 test cases generated for testing LLAIS for all the three networks.

Name of ne	**Hailfinder network**	
Hellinger's	**Approx. imp**	**Exact imp**
Minimum Error	0.0095	0.0075
Maximum Error	0.0147	0.0157
Mean	0.0118	0.0113
Median	0.0118	0.0111
Variance	1.99E-06	4.92E-06
Name of ne	**Win95pts network**	
Hellinger's	**Approx. imp**	**Exact imp**
Minimum Error	0.0084	0.0054
Maximum Error	0.0154	0.0178
Mean	0.0114	0.0095
Median	0.0114	0.0084
Variance	3.18E-06	1.03E-05
Name of ne	**Pathfinder network**	
Hellinger's	**Approx. imp**	**Exact imp**
Minimum Error	0.0168	0.0038
Maximum Error	0.1	0.0774
Mean	0.0403	0.0269
Median	0.0379	0.0313
Variance	6.05E-04	4.41E-04

5.2 Experiment Results for Improved LLAIS

After tuning the parameters as discussed in section 4, LLAIS shows considerable improvement in its accuracy and scalability with proper tuning of tunable parameters. Now the Improved LLAIS uses less number of samples and less updates in comparison to the Original LLAIS for giving posterior beliefs.

Fig. 7 shows the comparison of performance of Original LLAIS with Improved LLAIS and it can be seen that Improved LLAIS performs quite well showing good scalability on Hailfinder network.

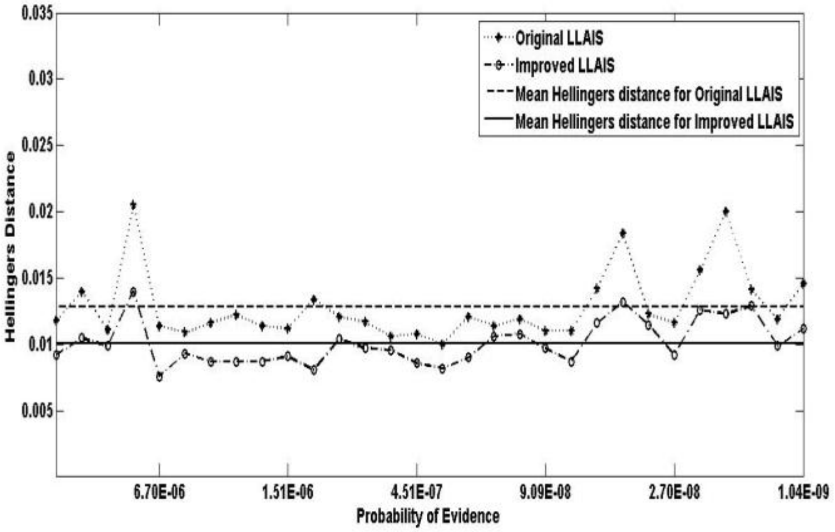

Figure 7: Performance comparison of Original LLAIS and Improved LLAIS for Hailfinder network. Hellinger's distance

Fig. 8 shows the comparison of performance of Original LLAIS with Improved LLAIS for Win95pts network and it can be seen in the graph that here also Improved LLAIS performed quite well with less errors as compared to the Original LLAIS.

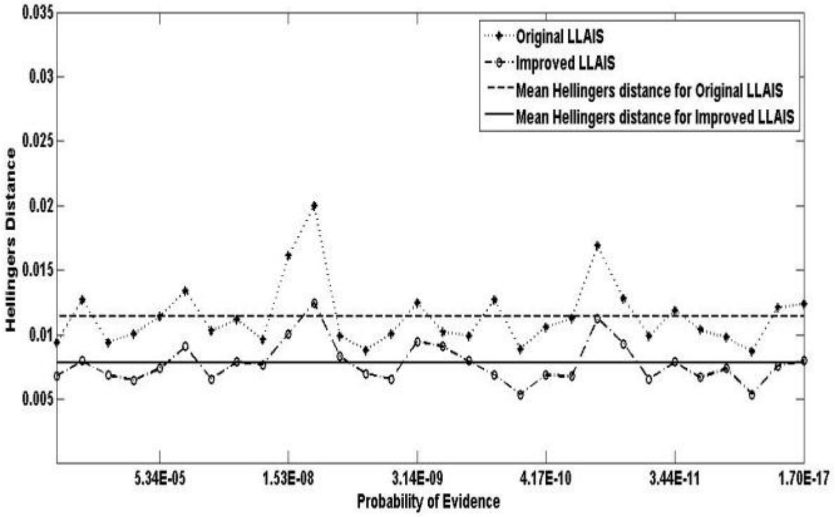

Figure 8: Performance comparison of Original LLAIS and Improved LLAIS for Win95pts network. Hellinger's distance for each of the 30 test cases plotted against $P(E)$

Fig 9 shows the comparison of performance of Improved LLAIS with Original LLAIS. The most extreme probabilities are found in this network, hence adjustments with threshold values played a key role in improving the performance; hence after tuning the parameters Improved LLAIS showed better performance in comparison to the original one for this network.

Table 3 shows the comparison of statistical results from all 30 test cases generated for Improved LLAIS and Original LLAIS.

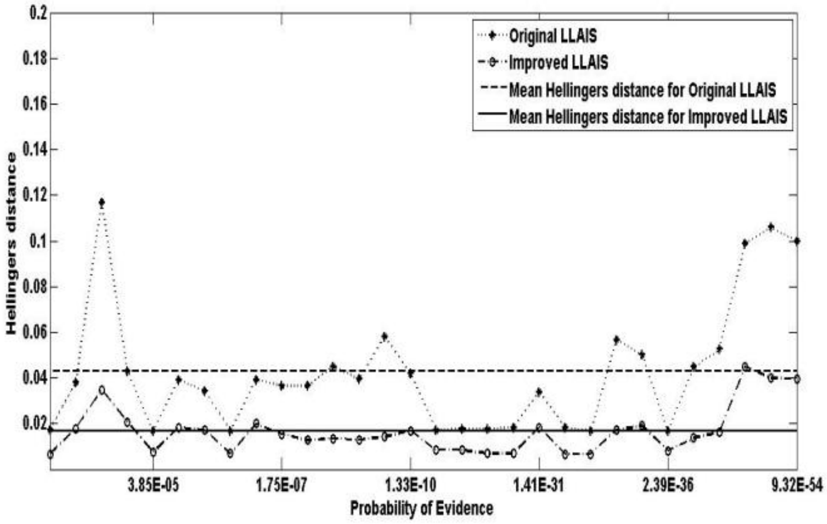

Figure 9: Performance comparison of original LLAIS and improved LLAIS for Pathfinder network. Hellinger's distance for each of the 30 test cases plotted against $P(E)$

Table 3: shows the comparison of results for original LLAIS and Improved LLAIS from all 30 test cases generated.

Name of ne	Hailfinder network	
Hellinger's	**Orignal**	**Improved**
Minimum Error	0.01	0.0076
Maximum Error	0.0205	0.014
Mean	0.0128	0.0101
Median	0.0119	0.0097
Variance	7.08E-06	2.73E-06
Name of ne	**Win95pts network**	
Hellinger's	**Orignal**	**Improved**
Minimum Error	0.0087	0.0054
Maximum Error	0.02	0.0125
Mean	0.0114	0.0078
Median	0.0105	0.0075
Variance	6.45E-06	2.50E-06
Name of ne	**Pathfinder network**	
Hellinger's	**Orignal**	**Improved**
Minimum Error	0.0168	0.0068
Maximum Error	0.117	0.0451
Mean	0.0427	0.0166
Median	0.0387	0.0149
Variance	7.80E-04	1.09E-04

6. Conclusion and Future Works

LLAIS is the extension of BN importance sampling to JTs. Since the preliminary testing of the algorithm was done only on smaller local-JT in LJF of 37 nodes, hence the scalability and reliability of the algorithm was questionable as the size of local-JTs may vary. From the experiments done, it can be concluded that LLAIS without parameters tuned performs quite well on local-JT of size 56 and 76 nodes but its performance deteriorates on 109 nodes network due to presence of extreme probabilities, once the parameters are tuned algorithm shows considerable improvement in its accuracy. It has been seen that learning time of the optimal importance function takes too long, so the choice of initial importance function $\Pr^0(X \setminus E)$ close to the optimal importance function can greatly affect the accuracy and convergence in the algorithm. As mentioned in [3], there is still one important question that remains unanswered how the local accuracy will affect the overall performance of the entire network. Further experiments are still to be done on the full scale MSBNs.

REFERENCES

[1] Karen H. Jin, ―Efficient probabilistic inference algorithms for cooperative Multi-agent Systems‖, Ph.D. dissertation, University of Windsor (Canada), 2010.

[2] Y.Xiang, Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach‖. Cambridge University Press, 2002.

[3] Karen H. Jin and Dan Wu, ―Local Importance Sampling in Multiply Sectioned Bayesian Networks‖, Florida Artificial Intelligence Research Society Conference, North America, May. 2010.

[4] Daphne Koller and Nir Friedman, Probabilistic Graphical Models-Principles and Techniques, MIT Press, 2009.

[5] Y.Xiang, ―Comparison of multiagent inference methods in Multiply Sectioned Bayesian Networks‖, International journal of approximate reasoning, vol. 33, pp.235-254, 2003.

[6] K.H.Jin and D.Wu, ―Marginal calibration in multi-agent probabilistic systems‖, In Proceedings of the 20th IEEE International conference on Tools with AI, 2008.

[7] J. Cheng and M. J. Druzdzel, ―BN-AIS: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks‖, Artificial Intelligence Research, vol.13, pp.155–188, 2000.

[8] C. Yuan, ―Importance Sampling for Bayesian Networks: Principles, Algorithms, and Performance‖, Ph.D. dissertation, University of Pittsburgh, 2006.