IPE and L2U: Approximate Algorithms for Credal Networks

Jaime S. Ide and Fabio G. Cozman
Escola Politécnica, Universidade de São Paulo
Departamento de Engenharia Mecatronica, Decision Making Lab
Av. Prof. Mello Moraes, 2231 - CEP 05508900, São Paulo, SP - Brazil

Abstract. This paper presents two approximate algorithms for inference in graphical models for binary random variables and imprecise probability. Exact inference in such models is extremely challenging in multiply-connected graphs. We describe and implement two new approximate algorithms. The first one is the Iterated Partial Evaluation (IPE) algorithm, directly based on the Localized Partial Evaluation (LPE) technique. The second one is the Loopy 2U (L2U) algorithm, an extension of the popular loopy belief propagation algorithm employed in Bayesian network inference. Experiments show excellent performance for these algorithms.

1 Introduction

Graphical models associated with probabilities find widespread use in artificial intelligence, both to represent and to reason about uncertainty. In this context, *Bayesian networks* are the most popular tool [17]. A Bayesian network is a directed acyclic graph, where each node is associated with a random variable and each arc represents a direct probabilistic dependency. A Bayesian network encodes a single joint distribution over all variables in the network, and an inference is a computation of conditional probability for some event. Inference techniques are well developed — among them, the junction tree algorithm [13] and variable elimination [7] are quite efficient methods.

In this paper we focus on situations where beliefs cannot be cast as sharp numeric values; that is, we have imprecise and incomplete beliefs that do not constrain probability values up to a single real number. Instead we consider sets of probability measures as suitable representations for beliefs. We deal with graphical models that represent uncertainty through sets of probability measures — graphical models that are often referred to as *credal networks* [5, 10]. In a credal network, a collection of sets of probability measures is associated with a directed acyclic graph. Inference in a credal network is the computation of lower and upper bounds for the conditional probability of some event. The complexity of inference in credal networks is generally high (even for tree-like networks [18]), and approximate inference seems to be a natural solution for large networks [1].

This paper describes two new algorithms for approximate inference in large credal networks: the Iterated Partial Evaluation (IPE) and the Loopy 2U (L2U) algorithms. The first algorithm is an extension of *Localized Partial Evaluation* (LPE), an algorithm for inference in Bayesian networks [9]. The IPE algorithm produces an approximation by combining a sequence of partial evaluations; each one of these partial evaluations deals with a polytree-shaped network with binary variables, a particular type of credal network for which the *2U* algorithm produces inferences in polynomial time [10]. The second algorithm is an extension of *loopy belief propagation*, a popular technique for approximate inference in Bayesian networks [16]. Loopy belief propagation has had notable success for approximate inference in Bayesian networks. We show how loopy belief propagation can be applied to credal networks and demonstrate that the result is an excellent method for approximate inference. We note that both IPE and L2U are briefly sketched in a related paper [6]; in this paper we present a complete derivation and discussion.

Sections 2 and 3 reviews basic facts about credal networks and the 2U algorithm respectively. Sections 4 and 5 describe the IPE and L2U algorithms respectively; results and discussions are presented in Section 6. Section 7 contains concluding remarks and a discussion of future work.

2 Credal networks

In this section we present a few facts on credal networks (and their basic elements, *credal sets*); a more detailed discussion can be found elsewhere [5, 10].

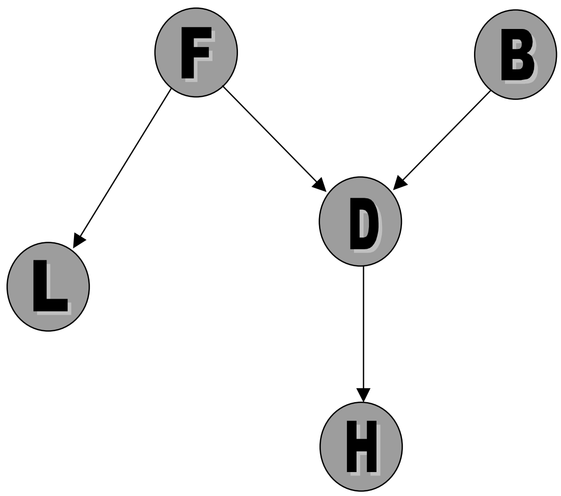
A convex set of probability distributions is called a *credal set* [14]. A credal set for variable X is denoted by $K(X)$; we assume that every variable is categorical and that every credal set has a finite number of vertices. A conditional credal set is a set of conditional distributions, obtained by applying Bayes rule to each distribution in a credal set of joint distributions [20]. Given a credal set $K(X)$ and a function $f(X)$, the *lower* and *upper* expectations of $f(X)$ are defined respectively as $\underline{E}[f(X)] = \min_{p(X) \in K(X)} E_p[f(X)]$ and $\overline{E}[f(X)] = \max_{p(X) \in K(X)} E_p[f(X)]$ (here $E_p[f(X)]$ indicates standard expectation). The *lower probability* and the *upper probability* of event A are defined respectively as $\underline{P}(A) = \min_{p(X) \in K(X)} P(A)$ and $\overline{P}(A) = \max_{p(X) \in K(X)} P(A)$. Lower and upper conditional probabilities for a variable X given an event E are defined accordingly:

$$\underline{P}(X = x|E) = \min_{p(X) \in K(X)} \frac{p(X = x, E)}{P(E)} \tag{1}$$

$$\overline{P}(X = x|E) = \max_{p(X) \in K(X)} \frac{p(X = x, E)}{P(E)}. \tag{2}$$

A *credal network* is a directed acyclic graph where each node of the graph is associated with a variable X_i and with a collection of conditional credal sets $K(X_i|\rho(X_i))$, where $\rho(X_i)$ denotes the parents of X_i in the graph (note that we have a conditional credal set for each value of $\rho(X_i)$). As an example, consider the "DogProblem" network (described by Charniak [4]). This network is associated with binary variables; suppose that the variables are associated with the credal sets indicated in Figure 1. If we observe that the dog is barking (evidence $H = true$), it is only possible to infer that $P(F = true|H = true) \in [0.18, 0.35]$.

Exact inference in general credal networks displays high complexity; even for polytree-shaped credal networks, inference is a NP-complete problem [18]. Such a situation has led to the development of several algorithms for approximate inference [3, 2, 1, 18, 19].



Node	Credal set
F	P(*F=true*) ∈ [*0.15 0.25*]
B	P(*B=true*) ∈ [*0.01 0.05*]
L	P(*L=true/F=true*) ∈ [*0.8 0.9*]
	P(*L=true/F=false*) ∈ [*0.01 0.1*]
D	P(*D=true/F=true,B=true*) ∈ [*0.2 0.3*]
	P(*D=true/F=true,B=false*) ∈ [*0.9 0.95*]
	P(*D=true/F=false,B=true*) ∈ [*0.05 0.1*]
	P(*D=true/F=false,B=false*) ∈ [*0.6 0.7*]
H	P(*H=true/D=true*) ∈ [*0.6 0.7*]
	P(*H=true/D=false*) ∈ [*0.01 0.05*]

Figure 1: Example of credal network: the DogProblem network, here associated with binary variables and collections of credal sets. F:family-out, B:bowel-problem, L:light-on, D:dog-problem and H:hear-bark.

3 The 2U algorithm

In this paper, we present algorithms for inference in general credal networks with *binary* variables.[1] The focus on binary variables is justified given the importance of such variables in qualitative reasoning, probabilistic logic and related fields [15]. Given that we rely critically on the 2U algorithm, in this section we present a detailed account of this procedure.

The 2U algorithms is an exact interval propagation algorithms for polytrees with binary variables [10]. This algorithm is a extension of Pearl's belief propagation [17] to probability intervals.

In Pearl's propagation, each node X computes a belief $Bel(x) = P(X = x|E) = \alpha\pi(X)\lambda(X)$, combining messages from its children (λ^X) and parents (π_X), where E denotes the set of observed nodes and α is a normalization constant [17]. To extend this algorithm to credal sets, it is necessary to absorb the constants α and β in order to compute $p(X|E)$ by means of composition of independent terms [10]. Hence new message definitions and formulas are derived as listed below:

[1]We should note that several ideas in this paper are applicable to non-binary networks, even though actual implementation leads to other complexity issues.

$$P(X=x|E) = \left(1+\left(\frac{1}{\pi(x)}-1\right)\frac{1}{\Lambda^X}\right)^{-1}, \tag{3}$$

$$\pi(x) = \sum_U p(x|U)\prod_i \pi_X(U_i), \tag{4}$$

$$\Lambda^X = \prod_j \Lambda^X_{Y_j}, \tag{5}$$

$$\pi_{Y_j}(x) = \left(1+\left(\frac{1}{\pi(x)}-1\right)\frac{1}{\prod_{k\neq j}\Lambda^X_{Y_j}}\right)^{-1}, \tag{6}$$

$$\Lambda^{U_i}_X = \frac{\rho(x|u_i)+(\Lambda^X-1)^{-1}}{\rho(x|\overline{u}_i)+(\Lambda^X-1)^{-1}}, \tag{7}$$

$$\rho(X|U_i) = \sum_{U_k\neq U_i} p(X|U)\prod_{k\neq i}\pi_X(U_k). \tag{8}$$

In Equation (3), a new quantity $\Lambda^X = \frac{\lambda(\overline{x})}{\lambda(x)}$ is defined (note that λ messages from Pearl's belief propagation are replaced by Λ). The π messages are computed using similar expressions (Equation (4)). Messages $\lambda^X_{Y_j}$ from children are replaced by $\Lambda^X_{Y_j}$. Messages sent to parents $\pi_{Y_j}(x)$ have new formulas (Equation (6)) as a function of Λ, and messages sent to children are written as functions of Λ and ρ (Equations (7) and (8)).

The 2U algorithm modifies the messages (3)-(8) to account for probability intervals (note that probability intervals are complete summaries of credal sets for binary variables):

$$\underline{P}(X=x|E) = \left(1+\left(\frac{1}{\underline{\pi}(x)}-1\right)\frac{1}{\underline{\Lambda}^X}\right)^{-1}, \tag{9}$$

$$\overline{P}(X=x|E) = \left(1+\left(\frac{1}{\overline{\pi}(x)}-1\right)\frac{1}{\overline{\Lambda}^X}\right)^{-1}, \tag{10}$$

$$\underline{\pi}(x) = \min_{\substack{j\,\in\,\{1,\ldots,n\}\\ \pi_X(u_j)\,\in\,\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \sum_U \underline{p}(x|U)\prod_i \pi_X(U_i), \tag{11}$$

$$\overline{\pi}(x) = \max_{\substack{j\,\in\,\{1,\ldots,n\}\\ \pi_X(u_j)\,\in\,\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \sum_U \overline{p}(x|U)\prod_i \pi_X(U_i), \tag{12}$$

Table 1: Extreme values for the subproblem of Equations (15) and (16).

	Different formulas depending on value of Λ^X	
	$\widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)$	$\overline{\widehat{\Lambda}}_X^{U_i}(\Lambda^X)$
$\Lambda^X < 1$	$\frac{\overline{\widehat{\rho}}(x\|u_i)+(\Lambda^X-1)^{-1}}{\underline{\widehat{\rho}}(x\|\overline{u}_i)+(\Lambda^X-1)^{-1}}$	$\frac{\underline{\underline{\widehat{\rho}}}(x\|u_i)+(\Lambda^X-1)^{-1}}{\overline{\widehat{\rho}}(x\|\overline{u}_i)+(\Lambda^X-1)^{-1}}$
$\Lambda^X = 1$	1	1
$\Lambda^X > 1$	$\frac{\underline{\underline{\widehat{\rho}}}(x\|u_i)+(\Lambda^X-1)^{-1}}{\overline{\widehat{\rho}}(x\|\overline{u}_i)+(\Lambda^X-1)^{-1}}$	$\frac{\overline{\widehat{\rho}}(x\|u_i)+(\Lambda^X-1)^{-1}}{\underline{\widehat{\rho}}(x\|\overline{u}_i)+(\Lambda^X-1)^{-1}}$

$$\underline{\Lambda}^X = \prod_j \underline{\Lambda}^X_{Y_j}, \tag{13}$$

$$\overline{\Lambda}^X = \prod_j \overline{\Lambda}^X_{Y_j}, \tag{14}$$

$$\underline{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\underline{\pi}(x)} - 1\right) \frac{1}{\prod_{k \neq j} \underline{\Lambda}^X_{Y_k}}\right)^{-1}, \tag{15}$$

$$\overline{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\overline{\pi}(x)} - 1\right) \frac{1}{\prod_{k \neq j} \overline{\Lambda}^X_{Y_k}}\right)^{-1}, \tag{16}$$

Equations (15) and (16) require the values:

$$\underline{\Lambda}_X^{U_i} = \min_{\substack{j \in \{1,\dots,n\},\, j \neq i \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \overline{\pi}_X(u_j)\}}} \left(\min_{\Lambda^X \in \{\underline{\Lambda}^X, \overline{\Lambda}^X\}} \widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X) \right),$$

$$\overline{\Lambda}_X^{U_i} = \max_{\substack{j \in \{1,\dots,n\},\, j \neq i \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \overline{\pi}_X(u_j)\}}} \left(\max_{\Lambda^X \in \{\underline{\Lambda}^X, \overline{\Lambda}^X\}} \overline{\widehat{\Lambda}}_X^{U_i}(\Lambda^X) \right),$$

where $\widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)$ and $\overline{\widehat{\Lambda}}_X^{U_i}(\Lambda^X)$ are computed according to Table 1.

For each node, through Equations (11)-(16), lower and upper messages are computed, basically following the structure of Pearl's propagation. Any node produces a local computation and the global computation is concluded by updating all nodes in sequence.

4 L2U: Loopy 2U

A popular algorithm for inference in Bayesian network is loopy propagation [16]. Loopy propagation applies Pearl's propagation to multiply connected networks. Here we describe a "loopy" variant of the 2U algorithm for multiply connected binary credal networks. First, a sequence S of nodes is randomly chosen, such that every node that is relevant to the inference

L2U: Loopy 2U algorithm
01. Load the network;
02. Initialize nodes without parents and evidence nodes.
03. Choose arbitrarily a sequence S being on a path in the network;
04. Repeat until the values converge or a time limit is reached:
05. for each node X of the sequence S, compute:
06. $\underline{\pi}^{(t+1)}(x)$ and $\overline{\pi}^{(t+1)}(x)$ given $\underline{p}(x|U)$, $\overline{p}(x|U)$ and messages from its parents $\pi_X^{(t)}(U_i)$;
07. $\underline{\Lambda}^{(t+1)X}$ and $\overline{\Lambda}^{(t+1)X}$ given messages from its children $\underline{\Lambda}_{Y_i}^{(t)X}$;
08. messages to be sent to its children $\underline{\pi}_{Y_j}^{(t+1)}(x)$ and $\overline{\pi}_{Y_j}^{(t+1)}(x)$ given $\underline{\pi}^{(t+1)}(x)$ and $\overline{\pi}^{(t+1)}(x)$, and messages from other children $\underline{\Lambda}_{Y_k}^{(t)X}$ and $\overline{\Lambda}_{Y_k}^{(t)X}$;
09. messages to be sent to its parents $\underline{\Lambda}_X^{(t+1)U_i}$ and $\overline{\Lambda}_X^{(t+1)U_i}$, given $p(X|U)$ given $\underline{\Lambda}^{(t+1)X}$, $\overline{\Lambda}^{(t+1)X}$, and messages from other parents $\underline{\pi}_X^{(t)}(U_k)$ and $\overline{\pi}_X^{(t)}(U_k)$;
10. Compute the queried nodes belief intervals $\underline{P}(X = x_q|E)$ and $\overline{P}(X = x_q|E)$, given $\underline{\pi}(x_q)$, $\overline{\pi}(x_q)$, and $\underline{\Lambda}^{X_q}$, $\overline{\Lambda}^{X_q}$.

Figure 2: The L2U algorithm.

is in S. Initialization of variables and messages follows the same steps used in the 2U algorithm. Then nodes are repeatedly updated following the sequence S. Iterations are indexed by i, which starts at 1, and updates are repeated until convergence of probabilities is observed or until a maximum number of iterations is reached. The algorithm is presented at Figure 2.

The whole algorithm follows the loopy propagation scheme, but instead of messages from Pearl's propagation algorithm, here we use interval messages from the 2U algorithm. As in Pearl's algorithm, nodes without parents and observed nodes are initialized (line 02), but here with some modifications given the new formulas for messages. If a node X is observed and set to value x, then $\Lambda^X = \lambda(x)/\lambda(\underline{x}) = 1/0 = \infty$; and if it is observed to $\underline{x}$, then $\Lambda^X = \lambda(x)/\lambda(\underline{x}) = 0/1 = 0$. In lines 06 and 07 messages are updated given the messages from parents and children of node X (Equations (11)-(14)), and in lines 08 and 09 messages to be sent to its parents and children (Equations (13)-(16)) are updated. Computation of Equations (9) and (10) for each node require a search within 2^n elements, where n is the number of parents. Computation of Equations (15) and (16) require a search within 2^{n-1} elements. Therefore, the overall complexity of L2U algorithm is $O(k \times 2^{2n_{max}})$ in the worst case, where k is the number of iterations and n_{max} is the maximum number of parents that a node can have.

5 Iterated Partial Evaluation (IPE)

Draper and Hank's Localized Partial Evaluation (LPE) algorithm produces approximate inferences by "cutting" local parts of a network and running interval-based inferences in a selected sub-network [9]. Our proposal is to compute inferences for parts of a network that form a polytree. We thus select a $cutset$ [17] of the network; we then create a polytree by cutting arcs from the variables in the cutset — these are the *missing arcs* in localized partial evaluation [9]. Figure 3 illustrates the process. LPE algorithm is then run on the polytree, but note that in binary networks we can use the 2U algorithm for these interval-based infer-

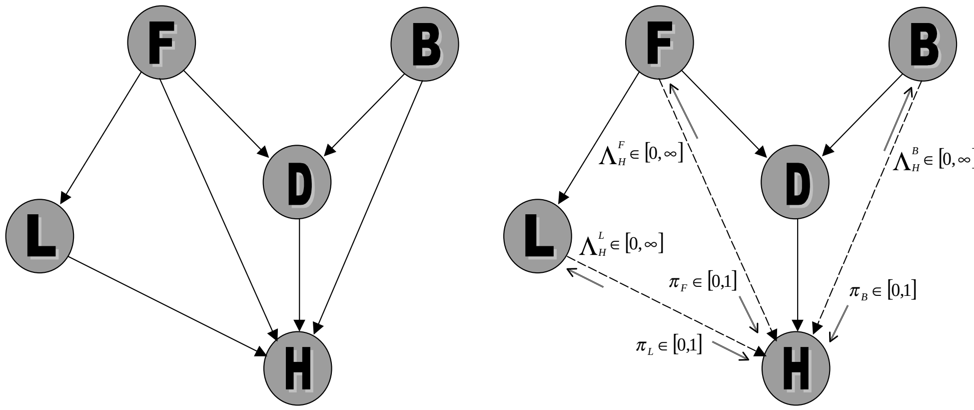


Figure 3: An example of *missing arcs* (dashed lines). Left: the multi-connected network; Right: the messages over the missing arcs.

IPE: Iterated Partial Evaluation algorithm
01. Load the network N;
02. Initialize nodes without parents and evidence nodes E.
03. Repeat i times:
04. find a $cutset$ and construct a polytree $N^*_{p_i}$, replacing some arcs by missing arcs;
05. for each polytree $N^*_{p_i}$, repeat 2U propagation and compute $Bel_{N^*_{p_i}}(x)$ of each variable X.
06. Obtain the intersection of $Bel_{N^*_{p_i}}(x)$ and compute $Bel_{N^*_p}(x)$.

Figure 4: The IPE algorithm.

ences. Using the 2U algorithm as the inference engine and in polynomial time, we obtain an approximate probability interval for any node in the credal network. The idea of the IPE algorithm is to repeat this process for several cutsets and to return the intersection of the resulting intervals.

The IPE algorithm is described in Figure 4. Denote by N the original credal network and by N^*_p a polytree obtained by deletion of missing arcs (given a cutset). In line 02, messages are initialized as in the L2U algorithm. In line 04, messages of missing arcs are set as follows: messages to be sent to its children, $\underline{\pi}_{Y_j}(x) = 0$ and $\overline{\pi}_{Y_j}(x) = 1$; and messages from other children $\underline{\Lambda}^X_{Y_k} = 0$ and $\overline{\Lambda}^X_{Y_k} = \infty$. Messages of missing arcs are not updated. We repeat lines 04 and 05 i times, updating messages of each variable and for each variable, we compute $Bel_{N^*_{p_i}}(x)$. Line 06 produces the intersection of intervals $Bel_{N^*_{p_i}}(x)$ obtained in each iteration i; thus we get at the end an interval $Bel_{N^*_p}(x)$. Consider that interval $Bel_N(x)$ has the extremes $\underline{P}(X = x|E)$ and $\overline{P}(X = x|E)$; then:

Theorem 1. *The probability interval produced by the IPE algorithm, $Bel_{N^*_p}(x)$, contains the exact interval $Bel_N(x)$ requested by the inference.*

Sketch of proof. The interval of each run of LPE, for each set of missing arcs, contains the point probability of an original Bayesian network defined by the credal network [8] — in credal networks, we know that the extreme values of $Bel_N(x)$ ($\underline{P}(X = x|E)$ and $\overline{P}(X = x|E)$) are computed from one combination of the local credal sets [10]. Then, the

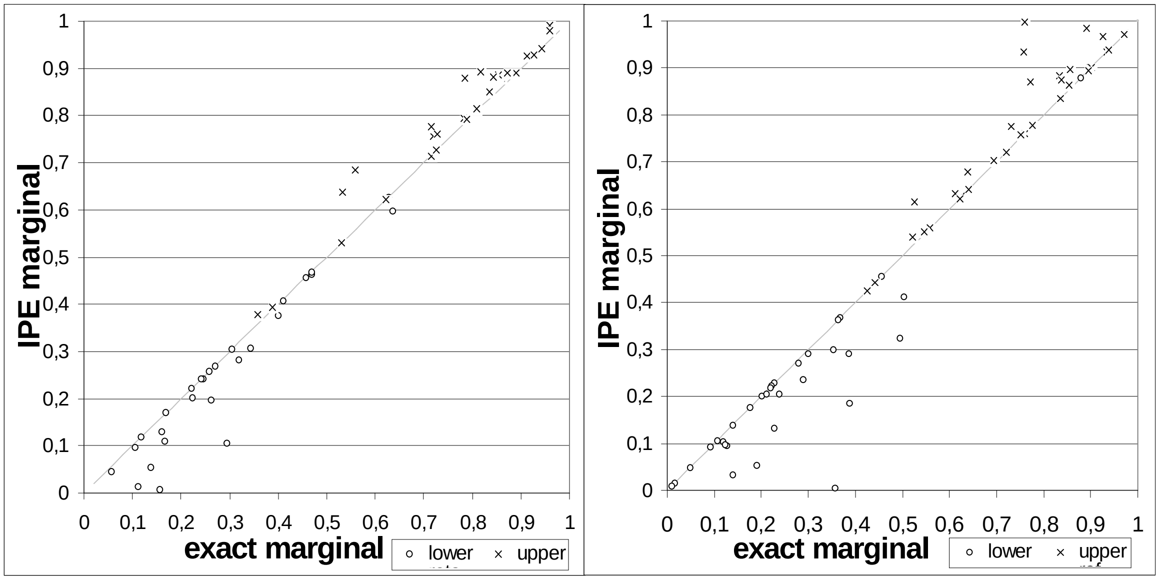


Figure 5: Experiments: (a) Pyramid network, with 20 variables, no evidence, 100 iterations and MSE=5,1%; (b) ”Binarized” Alarm network, with 37 variables, no evidence, 100 iterations and MSE=7,2%.

interval of all point probability combinations, $Bel_{N^*_{p_i}}(x)$, always contains the extreme values of $Bel_N(x)$, and the interval $Bel_{N^*_p}(x)$ obtained from intersection of all approximate intervals contains the exact interval. QED

6 Tests and results

We have implemented the IPE algorithm and run experiments in the network topologies employed by Murphy [16] to test loopy propagation: the Pyramid and Alarm networks. The Pyramid network is a multilayered graph associated with binary variables and local connections among layers. The Alarm network is a classic model used in medical diagnostic; we set all variables to binary values, so as to run IPE algorithm. For both networks, we generated several realizations of random, uniformly distributed conditional probability tables [12, 11]. Results can be viewed in Figure 5; most inferences are quite accurate, with mean square error (MSE) of 5% for Pyramid and 7.2% for Alarm.

The L2U algorithm was also implemented and tests were conducted in the same networks used to test the IPE algorithm. The L2U algorithm converges after 4 iterations in the Pyramid network, and after 9 iterations in ”binarized” Alarm network. The mean square error (MSE) of several approximate inferences was only 1.3% for both networks; these results can be viewed in Figure 6. It should be noted that L2U generally produces approximate inferences quite quickly: inferences (in all variables) for the ”binarized” Alarm network were produced in less than one second in a Pentium computer.

We also performed tests in very large and connected credal networks, and we obtained surprising results. For a very connected random network with 50 nodes, binary variables and induced width of 10, we obtained convergence with L2U in 13 iterations (about 8 minutes of processing time).[2]

[2]There is no way to quantify the accuracy of this result at this point, as there is no exact algorithm that can handle such networks.

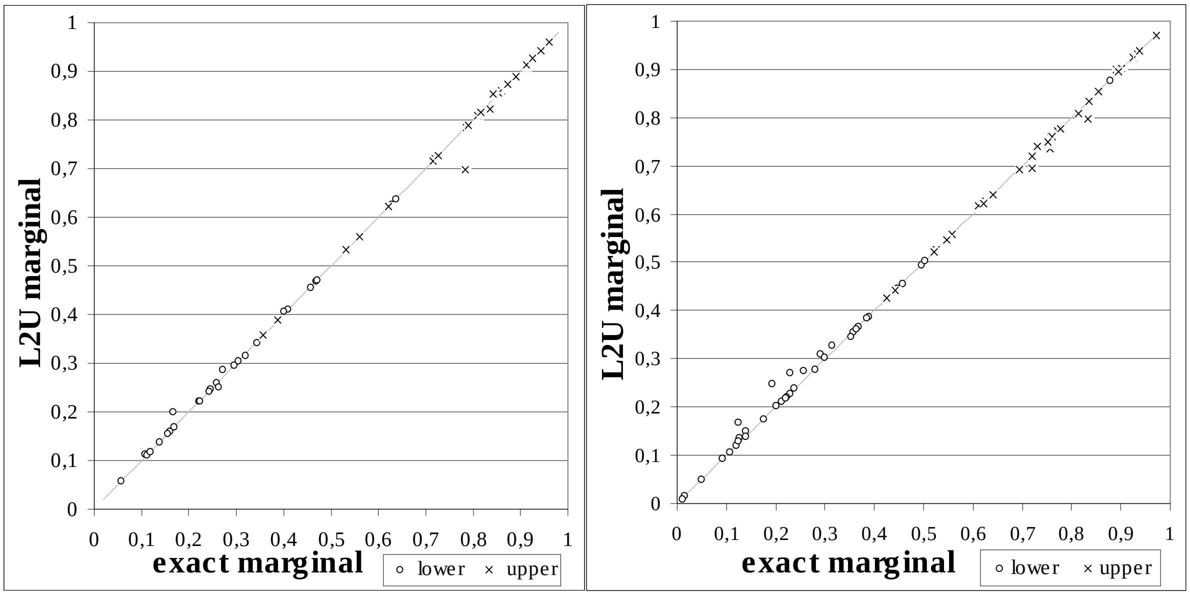


Figure 6: Experiments: (a) Pyramid network, with 20 variables, no evidence, and 4 iterations;(b) ”Binarized” Alarm network, with 37 variables, no evidence, and 9 iterations.

Overall, the L2U algorithm is fast and returns satisfactory results, with low errors. However it displays the same disadvantage of all loopy propagation algorithms: there are no theoretical guarantees about convergence. On other hand, the IPE algorithm gives bounds that always contain the right answer.

7 Conclusion

This paper describes in detail two new algorithms for approximate inference in credal networks: the IPE and L2U algorithms. Their implementations are limited to networks with binary variables as they make critical use of the polynomial character of the 2U algorithm. However, the central ideas in IPE and L2U are not limited to binary networks, and could be extended, in future work, to more general kinds of networks.

The L2U algorithm is particularly promising for practical application, even though a solid convergence analysis is missing at this point. Clearly several challenges are yet to be overcome, but it seems that the algorithms presented here are viable methods for fast inference in large credal networks.

Acknowledgements

The first author was supported by FAPESP grant 02/08982-2. We thank Denise Draper for providing a copy of her thesis. This work was (partially) developed in collaboration with HP Brazil R&D. The second author has also been partially supported by CNPq (through grant 3000183/98-4).

References

[1] A. Cano and S. Moral. Using probability trees to compute marginals with imprecise probabilities. *Int. Journal of Approximate Reasoning*, 29:1–46, 2002.

[2] A. Cano and S. Moral. A genetic algorithm to approximate convex sets of probabilities. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2:859–864, 1996.

[3] J. Cano, M. Delgado, and S. Moral. An axiomatic framework for propagating uncertainty in directed acyclic networks. *Int. Journal of Approximate Reasoning*, 8:253–280, 1993.

[4] E. Charniak. Bayesian Networks without tears. In *AI Magazine*, Vol. 12, pp. 50-63, 1991.

[5] F. G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[6] F. G. Cozman, C. P. de Campos, J. S. Ide and J. C. F. Rocha Propositional and Relational Bayesian Networks Associated with Imprecise and Qualitative Probabilistic Assessments In *Conference on Uncertainty in Artificial Intelligence*, 2004 (to appear).

[7] R. Dechter. Bucket elimination: An unifying framework for probabilistic inference. In *Conf. on Uncertainty in Artificial Intelligence (UAI-96)*, pages 211–219, San Francisco, August 1–4 1996. Morgan Kaufmann Publishers.

[8] D. L. Draper. *Localized Partial Evaluation of Belief Networks*. PhD thesis, Dept. of Computer Science, University of Washington, Washington,WA, 1995.

[9] D. L. Draper and S. Hanks. Localized partial evaluation of belief networks. In *Conf. on Uncertainty in Artificial Intelligence*, Seattle, WA, 1994. Morgan Kaufmann.

[10] E. Fagiuoli and M. Zaffalon. 2U: an exact interval propagation algorithms for polytrees with binary variables. *Artificial Intelligence*, 106:77–107, 1998.

[11] J. S. Ide, F. G. Cozman and F. T. Ramos *Generation of Random Bayesian Networks with Constraints on Induced Width, with Applications to the Average Analysis od d-Connectivity, Quasi-random Sampling, and Loopy Propagation*. University of Sao Paulo, Technical Report, June, 2003.

[12] J. S. Ide and F. G. Cozman. Random generation of Bayesian networks. In *Brazilian Symposium on Artificial Intelligence*. Springer-Verlag, 2002.

[13] F. V. Jensen, K. G. Olensen, S. K. Andersen. An algebra of Bayesian belief universes for knowledge-based systems. In *Networks*, Vol. 20, pp. 637-659, 1990.

[14] I. Levi. *The Enterprise of Knowledge*. MIT Press, Cambridge, Massachusetts, 1980.

[15] T. Lukasiewicz. Probabilistic logic programming. *European Conf. on Artificial Intelligence*, pp. 388-392, 1998.

[16] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Conf. on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[17] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufman, 1988.

[18] J. C.F. Rocha and F G. Cozman. Inference with separately specified sets of probabilities in credal networks. In *Conf. on Uncertainty in Artificial Intelligence*, pages 430–437, San Francisco, CA, 2002. Morgan Kaufmann.

[19] B. Tessem. Interval probability propagation. *Int. Journal of Approximate Reasoning*, 7:95–120, 1992.

[20] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.