Epsilon-cutset effect in Bayesian networks of arbitrary topology

Bozhena Bidyuk
Department of Information and Computer Science
University Of California Irvine
Irvine, CA 92697-3425
bbidyuk@ics.uci.edu

Rina Dechter
Department of Information and Computer Science
University Of California Irvine
Irvine, CA 92697-3425
dechter@ics.uci.edu

Abstract

The paper investigates the behavior of iterative belief propagation algorithm (IBP) in Bayesian networks with loops. In multiply-connected network, IBP is only guaranteed to converge in linear time to the correct posterior marginals when evidence nodes form a loop-cutset. We propose an ϵ-cutset criteria that IBP will converge and compute posterior marginals close to correct when a single value in the domain of each loop-cutset node receives very strong support compared to other values thus producing an effect similar to the observed loop-cutset. We investigate the support for this criteria analytically and empirically and show that it is consistent with previous observations of IBP performance in multiply-connected networks.

1 Introduction

The paper investigates the correctness of iterative belief propagation (IBP) algorithm in Bayesian networks with loops. Pearl [10] proposed iterative belief propagation algorithm for singly connected Bayesian networks and demonstrated that algorithm converges in the number of iterations equals to the diameter of the network to the correct posterior values. Iterative belief propagation can be applied to networks with loops to derive approximate inference where exact methods such as bucket-elimination [2] and tree-clustering [9, 3] become impractical due to exponential growth in time and memory required as network width increases. In general, it does not always converge and does not produce correct posterior values for Bayesian networks with loops.

However, empirically it was demonstrated that IBP can be successfully applied to several classes of Bayesian networks with loops used in practical applications, especially for coding networks [11, 8, 4]. where it was shown to outperform variational decoder [4] and mini-bucket approximation algorithm [6]. It was also performs well on noisy-or networks (used in diagnostics) and pyramid networks (used in image recognition) [7].

We have only limited theoretical understanding of the behavior of IBP in networks with loops. Most of the results refer to the Bayesian networks with a single loop. Weiss [12] proved using the Markov network model that IBP always converges on a single-loop networks and defined the error in posterior marginals obtained by IBP as a function of eigenvalues of a matrix computed from the conditional probability tables of all the variables in a loop. He also established the correlation between the accuracy of posterior marginals computed by IBP and convergence rate. That is, the faster IBP convergence, the more accurate the posterior marginals are.

We investigate the accuracy of IBP in directed Bayesian networks with binary nodes as a function of loop size, CPT values, prior beliefs, and evidence support. Analytically, we derive an expression for the error value in the posterior belief of a sink node in a single-loop Bayesian network without evidence (section 5) and then extend our conclusions empirically to several classes of loopy networks with evidence (section 6). The general conclusion is that in a single-loop network accuracy of IBP improves as:

1. Prior beliefs for a root node(s) approach boundary values of 0 and 1.
2. Size of the loop increases.
3. Conditional probabilities of an allowed loop node X for different values of its parent $P(X = x_0|Pa(X))$ and $P(X = x_0|Pa(X) = 1)$ get closer.

For networks with multiple loops, we propose an ϵ-cutset criteria that IBP will converge and compute posterior marginals within specified error limit δ when ϵ-support is provided for loop-cutset nodes with a sufficiently small ϵ (section 3).

2 Background

Definition **2.1 (graph concepts)** *A* directed graph *is a pair $G = \{V, E\}$, where $V = \{X_1, ..., X_n\}$ is a set of nodes, or variables, and $E = \{(X_i, X_j)|X_i, X_j \in V\}$ is the set of edges. Given $(X_i, X_j) \in E$, X_i is called a* parent *of X_j, and X_j is called a* child *of X_i. The set of X_i's parents is denoted $pa(X_i)$, or pa_i, while the set of X_i's children is denoted $ch(X_i)$, or ch_i. The family of X_i includes X_i and its parents.*

The underlying graph G of a directed graph D is the undirected graph formed by ignoring the directions of the edges in D.

A node X in a directed graph D is called a root *if no edges are directed into X. A node X in a directed graph D is called a* leaf *if all of its adjacent edges are directed into X. A* cycle *in G is a path whose two end-points coincide. A* cycle-cutset *of undirected graph G is a set of vertices that contains at least one node in each cycle in G. A* loop *in D is a subgraph of D whose underlying graph is a cycle. A vertex v* *is a* sink *with respect to loop* $\mathcal{L}$ *if the two edges adjacent to* v *in* $\mathcal{L}$ *are directed into* v. *A vertex that is not a sink with respect to a loop* $\mathcal{L}$ *is called an* allowed *vertex with respect to* $\mathcal{L}$. *A* loop-cutset *of a directed graph* D *is a set of vertices that contains at least one allowed vertex with respect to each loop in* D. *(We borrowed loop-cutset definition from [1]).*

A directed graph is acyclic if it has no directed *cycles. A graph is* singly connected *(also called a* polytree*), if its underlying undirected graph has no cycles. Otherwise, it is called* multiply connected.

DEFINITION **2.2 (belief networks)** *Let* $X = \{X_1, ..., X_n\}$ *be a set of random variables over multi-valued domains* $D_1, ..., D_n$. *A* belief network (BN) *is a pair* (G, P) *where* G *is a directed acyclic graph on* X *and* $P = \{P(X_i|pa_i)|i = 1, ..., n\}$ *is the set of conditional probability matrices associated with each* X_i. *An assignment* $(X_1 = x_1, ..., X_n = x_n)$ *can be abbreviated as* $x = (x_1, ..., x_n)$. *The BN represents a joint probability distribution* $P(x_1, ..., x_n) = \Pi_{i=1}^n P(x_i|x_{pa(X_i)})$, *where* x_S *is the projection of vector* x *on a subset of variables* S. *An evidence* E *is an instantiated subset of variables.*

DEFINITION **2.3 (d-separation)** *If* X, Y, *and* Z *are three disjoint subsets of nodes in a DAG* G, *then* Z *is said to d-separate* X *from* Y, *denoted* $d(X, Z, Y)_G$, *if and only if there is not path from a node in* X *to a node in* Y *along which the following two conditions hold: (1) every node with converging arrows either is or has a descendant in* Z, *and (2) every other node is outside* Z. *A path satisfying the conditions above is to be* active*; otherwise it is said to be* blocked *(by* Z*). By path we mean a sequence of consecutive edges (of any directionality) in the DAG.*

3 Iterative Belief Propagation Algorithm

Iterative Belief Propagation (IBP) computes belief $BEL(x) = P(X = x|E)$, where E is observed evidence, for every variable X in the network. It applies Pearl's belief propagation algorithm [10], developed for singly-connected networks, to a multiply-connected networks, ignoring cycles. Belief is propagated by sending messages between the nodes:

During each iteration $(t+1)$, each node X sends *causal* support messages $\pi_{Y_j}^{(t+1)}(x)$ to each child Y_j:

$$\pi_{Y_j}^{(t+1)}(x) = \alpha \lambda_X(x) \prod_{k \neq j} \lambda_{Y_k}^t(x) \pi^{(t)}(X) \tag{1}$$

and *diagnostic* support messages $\lambda_x^t(u_i)$ to each parent u_i:

$$\lambda_X^{(t+1)}(u_i) = \beta \sum_X \lambda^{(t)}(x) \sum_{u_k, k \neq i} P(x|u) \prod_{k \neq i} \pi_X^{(t)}(u_k) \tag{2}$$

where

$$\lambda^{(t)}(x) = \lambda_X(x) \prod_j \lambda_{Y_j}^{(t)}(x) \tag{3}$$

$$\pi^{(t)}(x) = \sum_u P(x|u) \prod_i \pi_X^{(t)}(u_i) \tag{4}$$

In equations 1 and 2, α and β are normalization constants such that $\sum_{u_i} \lambda_X(u_i) = 1$ and $\sum_X \pi_{Y_j}(X) = 1$. Message $\lambda_X(X)$ is introduced to incorporate evidence information into the equation (similar to [7]). If node X is not observed, $\forall x_k \in D_X, \lambda_X(X = x_k) = 1$. If node X is observed and x_e is the evidence value, then $\forall x_k \in D_x, x_k \neq x_e, \lambda_X(X = x_k) = 0$ and $\lambda_X(x = x_e) = 1$.

The posterior belief is computed for each node X by combining $\pi_X(u_i)$ messages received from its parents u_i and $\lambda_{Y_j}(X)$ messages received from its children:

$$BEL(x) = \alpha \lambda^{(t)}(X) \pi^{(t)}(X) \tag{5}$$

Causal supports from all parents and diagnostic support from all children are combined into vectors π_x and λ_x, respectively.

Normalization of π and λ messages is recommended to avoid numerical underflow although it does not affect the computation of the posterior beliefs. We can show that $P^n(X|E) = K * P^{un}(X|E)$ where $P^n(X|E)$ is a belief in X computed using normalized messages, $P^{un}(X|E)$ without normalizing π and λ messages, and K is constant independent on the value of X. The constant K is going to disappear after we normalize the belief in X (see appendix 1).

An *activation schedule* (variable ordering) A specifies the order in which the nodes are processed (activated). After all nodes are processed, the next iteration of belief propagation begins, updating the messages computed during the previous iteration. Algorithm IBP(n) stops after n iterations.

4 IBP, loop-cutset and irrelevant subnetworks

In this section, we demonstrate that IBP *automatically* exploits conditional independence introduced by observed nodes and automatically ignores irrelevant portions of the network that contain no evidence.

Utilizing the notion of conditional independence, we can often reduce a seemingly complex Bayesian network to a singly-connected one. It is well-known that if evidence nodes form a loop-cutset, then we can transform multiply-connected Bayesian network to an equivalent singly-connected network which can be solved by belief propagation, leading to the loop-cutset conditioning method [10].

Also, unobserved nodes that have only unobserved descendents are irrelevant to the beliefs of the remaining nodes and therefore, processing can be restricted to the relevant subgraphs.

Combining the above two properties, it is clear that if evidence nodes constitute a loop-cutset of a relevant subgraph of a query node X, then its posterior belief can be computed by applying belief-propagation only to the relevant subgraph which can be *transformed* into a singly-connected network.

We show in subsections 4.1 and 4.2 that IBP exploits the above two properties, of observed and unobserved nodes, automatically, without requiring any outside action for network transformation. As a result, the correctness and convergence of IBP on node X in a multiply-connected Bayesian network will be determined by the structure of the relevant subgraph of node X as opposed to the structure of a complete network. If relevant subnetwork of node X is singly-connected, IBP will converge to correct posterior marginals for node X.

While these conclusions are fairly straightforward, we belief they deserve to be stated towards the understanding of IBP's boundaries as an approximate scheme.

In section 4.3, we discuss how strong support for one value of a node (for example, from its observed children) may lead to the approximation of the conditional independence effectively weakening dependence between the node's children and parents. We further argue that in a multiply-connected Bayesian network, as each loop-cutset node receives stronger support for one value, the closer the approximation of loop-cutset condition becomes leading to the improvements in IBP convergence and accuracy.

4.1 Evidence nodes

An observed node X in a Bayesian network G blocks the path between its parents and its children as defined in d-separation criteria. In other words, it creates conditional independence between its parents and its children.

In this section we establish that IBP automatically exploits evidence nodes blocking information flow between their parents and their children. Namely, messages that observed node X sends to its children are independent from any messages that node X receives. Message that observed node X sends to its parents are independent from messages node X receives from its children.

LEMMA **4.1 (Information flow blocking)** *Let X be an observed node in a Bayesian network G. Then for any child Y_j of node X, the $BEL(Y_j)$ computed by IBP is independent from the messages that X receives from its parents U_i or the messages that node X receives from its other children $Y_k, k \neq j$.* □

Lemma 4.1 allows us to understand fully the behavior of IBP in a Bayesian network where observed nodes form a loop-cutset.

THEOREM **4.1 (IBP on loop-cutset)** *If evidence nodes constitute a loop-cutset, then IBP converges to the correct posteriors in linear time.* □

The proof for lemma 4.1 and theorem 4.1 is given in appendix 2.

4.2 Irrelevant nodes

Unobserved nodes that have only unobserved descendents are irrelevant to the beliefs of the remaining nodes and therefore, processing can be restricted to the relevant subgraphs. In IBP, this property is expressed by the fact that irrelevant nodes (that are not observed and do not have observed descendents) send diagnostic support messages that equally support each value in the domain of a parent and thus do not affect the computation of marginal posteriors of its parents.

LEMMA **4.2** *Let X be a node in a loopy Bayesian network G such that it is not observed and it does not have observed descendents and let G' be a subnetwork obtained by removing node X and its descendents from the network. Then for any $Y \in G'$, the posterior belief of Y as computed by IBP over G is identical to the posterior belief of Y computed by IBP applied to G' only.* □

We can immediately conclude that in a loopy network without evidence, IBP will always converge after 1 iteration because only propagation of π messages affects the computation of posterior beliefs and π messages do not change. Also in that case, IBP converges to the correct marginals for any node whose parents do not have common ancestors. This is because the relevant subnetwork that contains only the node and its ancestors is singly-connected. Therefore, by lemma 4.2 they are the same as the posterior marginals computed by applying IBP to the complete network. In summary,

THEOREM **4.2 (Irrelevant unobserved nodes)** *Let G be a Bayesian network and let G' be a network obtained by recursively eliminating all its unobserved leaf nodes. If observed nodes in G constitute a loop-cutset of G', then for all the nodes in G', IBP applied to G converges to the correct posterior marginals. For a node outside G', IBP will converge (not necessarily to correct posteriors) only if it converges for all of its parents.* □

Lemma 4.2 and theorem 4.2 are proved in appendix 2.

An interesting consequence of theorem 4.2 is that IBP is guaranteed to converge for any Bayesian network that has no observed nodes or that has the observed root nodes only.

4.3 ϵ-cutset

In multiply-connected networks, IBP is only guaranteed to converge to correct posterior marginals for all nodes if evidence nodes form a loop-cutset. An observed loop-cutset node X effectively breaks the loop by stopping the flow information between its parents and its children.

Now, assume that node X is not observed but receives very strong support for one value in its domain, x', from prior beliefs $|1 - P(X = x')| < \epsilon$ and/or support messages from its children $|1 - \lambda_{Y_k}(X = x')| < \epsilon$ (assume that λ has been normalized). We will call it ϵ-support. Then $lim_{\epsilon \to 0} \lambda^t(X = x') \to 1$ generating effect similar to the observed nodes. Therefore, we make following proposition:

Proposition 1 (ϵ-cutset) *Let G be a multiply-connected Bayesian network. Let X_i, $i = 1, ..., m$, form a loop-cutset in G. Then, there exists such value $\epsilon_0 \in (0, 1)$ that if each loop-cutset node X_i receives an ϵ_0-support $|1 - P(X_i = x'_i)| < \epsilon_0$ or $|1 - \lambda_{Y_k}(X_i = x'_i)| < \epsilon_0$ for normalized $\lambda_{Y_k}(X_i)$, then IBP will converge and compute posterior marginals within specified error limit δ.* □

Proposition 1 extends the loop-cutset evidence influence on IBP to extreme but probabilistic evidence nodes. We support this hypothesis by several theorems on single loops (next section) and provide general validation empirically.

5 Single Loop Bayesian Network without Evidence

In this section, we will analyze the performance of belief propagation algorithm in a single-loop Bayesian network with binary nodes as shown in fig 1. Without evidence, the posterior marginals of all nodes except the sink node D will be computed correctly due to theorem 4.2. Thus, we focus here on computing the error

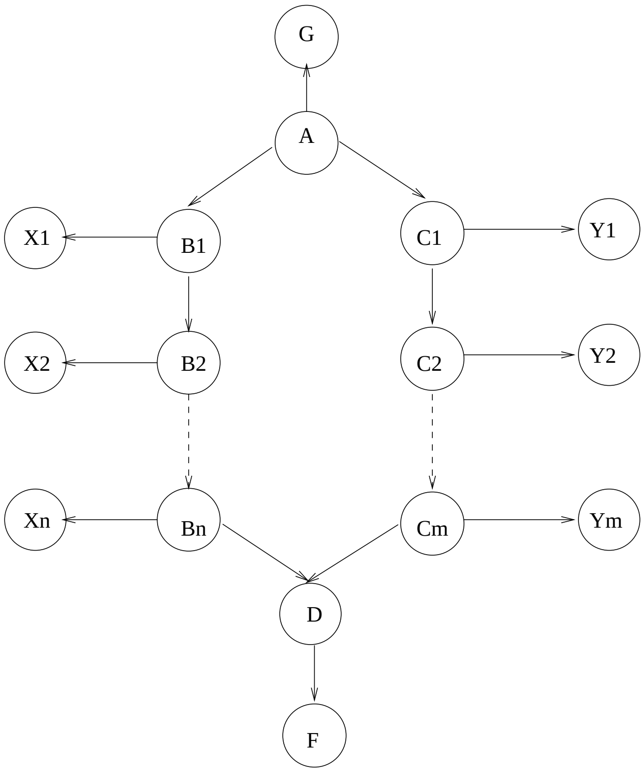

Figure 1: A single loop Bayesian network.

produced by IBP in the posterior marginal values of sink node D. Let $G(D)$ be correct posterior belief:

$$G(D) = \sum_{B_n, C_m} P(D|B_n, C_m) \sum_{A} P(B_n|A) P(C_m|A) P(A) \tag{6}$$

In a Bayesian network without observed nodes, all λ messages sent from children to parents provide same support for all parent values (λ will be vectors with all values set to 1). Therefore, only π messages contribute to the computation of the marginal beliefs. If we create activation schedule such that parents of a node are processed first, then one iteration of belief propagation is sufficient for convergence. The messages sent in future iterations would be identical to the messages sent during first iteration. Then, it is easy to derivethe posterior marginals of node D $G^*(D)$ computed by IBP:

$$G^*(D) = \alpha \sum_{B_n, C_m} P(D|B_n, C_m) \pi(B_n) \pi(C_m) =$$

$$= \alpha \sum_{B_n, C_m} P(D|B_n, C_m) (\sum_{A} P(B_n|A) P(A)) (\sum_{A} P(C_m|A) P(A)) \tag{7}$$

where α is a normalization constant. We can show that $\alpha = 1$ when neither D nor any of its descendents are observed. In other words, $G^*(D)$ does not require normalization.

THEOREM **5.1** *Let G be a single-loop Bayesian network as shown in figure 1 Assume all nodes are binary and there are no observed nodes in the loop. Let us define:*

$$P(A) = (\epsilon, 1-\epsilon)$$

$$\beta_0^i = P(B_i = 0|B_{i-1} = 0), \beta_1^i = P(B_i = 0|B_{i-1} = 1)$$
$$\gamma_0^j = P(C_j = 0|C_{j-1} = 0), \gamma_1^j = P(C_j = 0|C_{j-1} = 1)$$

Then the error $\delta = G^*(D) - G(D)$ in the posterior marginals of node D computed by Iterative Belief Propagation will be:

$$\delta = (\epsilon - \epsilon^2)((-1)^{B_n}(-1)^{C_m}\sum_{B_n, C_m} P(D|B_n, C_m))\prod_{i=1}^{n}(\beta_0^i - \beta_1^i)\prod_{j=1}^{m}(\gamma_0^j - \gamma_1^j) \quad (8)$$

We prove this theorem in appendix 3. Let $d_{ij} = P(D = 0|B_n = i, C_m = j)$. Then, for D=0:

$$\delta = \epsilon(1-\epsilon)(d_{00} - d_{01} - d_{10} + d_{11})\prod_{i=1,n}(\beta_0^i - \beta_1^i)\prod_{j=1,m}(\gamma_0^j - \gamma_1^j) \quad (9)$$

From equation 9, it is clear that the accuracy of IBP improves:

1. As prior beliefs for a root node(s) approach boundary values: $\lim_{\epsilon \to 0,1} \delta = 0$
2. As λ support messages from the children of allowed nodes in the loop approach boundary values: $\lim_{\lambda(B),\lambda(C),\lambda(A) \to 0,1} \delta = 0$
3. As number of nodes in a loop increases: $\lim_{n\to\infty, m\to\infty} \delta = 0$
4. As conditional probabilities for the same value of X in different rows get closer:
$$\lim_{\beta_0^k - \beta_1^k -> 0, \gamma_0^k - \gamma_1^k -> 0} \delta = 0 \quad (10)$$

It is also easy to see that when one of the allowed nodes is observed, $\delta = 0$. When node A is initialized, either $\epsilon = 0$ or $(1-\epsilon) = 0$. When one of the nodes B_i or C_j is observed, it is equivalent to having $\beta_0(B_i) = \beta_1(B_i) = 0|1$ or $\gamma_0(C_j) = \gamma_1(C_j) = 0|1$ yielding $\delta = 0$.

6 Empirical Results

In this section, we empirically investigate accuracy and convergence of IBP in networks with loops. In all experiments, unless specified otherwise, conditional probabilities were represented by noisy-or:

$$P(Child = 0|Parents) = e^{-\theta_0 - \sum_i \theta_i Parent_i} \quad (11)$$

where θ_0, the 'leak' term was fixed at 0.005 and individual noise factors θ_i were chosen uniformly from the range [0,1]. The number of iterations for approximate inference was fixed at 20. All experimental results are consistent with conclusions in section 5 and proposition 1.

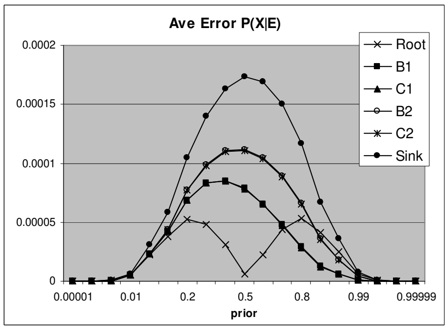

Figure 2: Average error in $P(X|E)$ is plotted against $P(A=0)$.

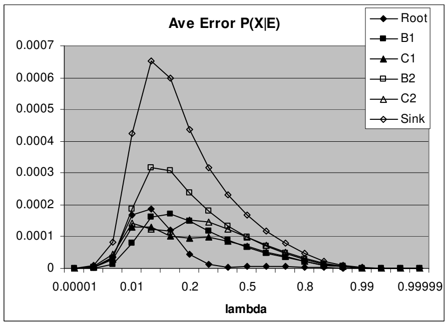

Figure 3: Average error in $P(X|E)$ is plotted against $\lambda_{Y_2}(C_2)$.

6.1 Single Loop

Single loop networks (see fig 1) containing 4, 6, 8, and 10 nodes were constructed. The child of a sink node was always observed. The average error value was averaged over 1000 instances. As the loop size increased, the maximum average error value decreased by the order of magnitude: 0.002 for a 4-node loop, 0.0002 for a 6-node loop, 0.00002 for a 8-node loop, and 0.000002 for a 10-node loop.

For all nodes and all loop sizes, the average error value rapidly decreased as $P(A)$ was approaching 0 and 1 as predicted analytically in equation 9 (results for a 6-node loop are presented in fig 2). For all nodes, except root node, the average error value peaked at $P(A) = 0.5$, as expected, since $\epsilon = 0.5$ is the maximum of the function $f(\epsilon) = \epsilon(1-\epsilon)$. The average error for a root node consistently had a minimum at $P(A) = 0.5$ which we cannot explain analytically at this point and plan to investigate in the future. For all nodes and all loop sizes, the average error was approaching zero as $\lambda_{Y_m}(C_m) \to 0, 1$ (results for a 6-node loop are presented in fig 3).

6.2 2-layer noisy-or networks

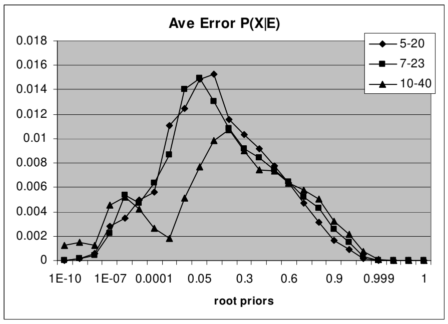

Figure 4: Average error in $P(X|E)$ is plotted against root node priors.

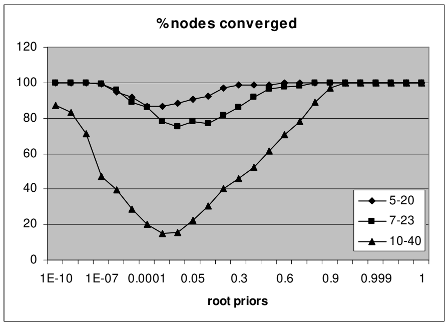

Figure 5: Percent of converged nodes is plotted against root node priors.

We measured the accuracy and convergence of IBP in 2-layer noisy-or networks.

Number of root nodes m and total number of nodes n was fixed in each test set (indexed $m-n$). Each leaf node Y_j was added to the list of children of root node U_i with probability 0.5. All leaf nodes were observed. Loop-cutset nodes were selected from root nodes using mga algorithm [1]. Results (in figures 4 and 5) were averaged over 100 instances. We have observed in our experiments that initially, as priors of loop-cutset nodes become lower, the convergence and accuracy of IBP worsen. This effect was previously reported by Murphy, Weiss, and Jordan [7] for 2-layer noisy-or networks with low root node priors. However, as priors of loop-cutset nodes continue to approach 0 and 1, the average error value approaches 0 and the number of converged nodes reaches 100%.

6.3 Random noisy-or networks

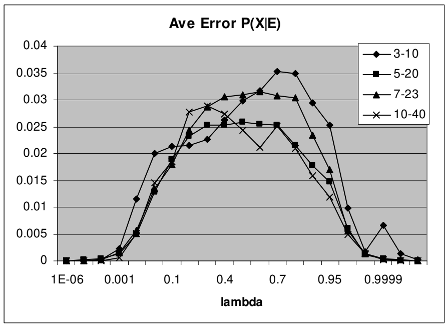

Figure 6: Average error in $P(X|E)$ is plotted against $\lambda_{Y_k}(X_k)$.

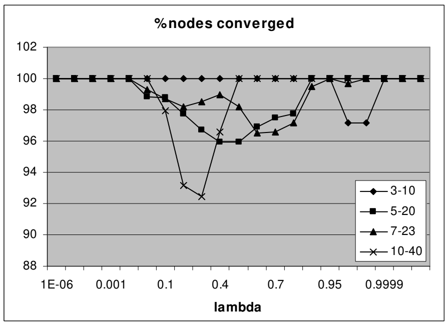

Figure 7: Percent of converged nodes is plotted against $\lambda_{Y_k}(X_k)$.

Random $m-n$ networks of size n were constructed by designating first m nodes as roots, last m nodes as leaves, and then adding each node X_i, $i > m$, to the list of children of node X_j, $j > i$, with probability 0.2. All leaf nodes were observed. Loop-cutset nodes were selected using mga algorithm proposed in [1]. For each loop-cutset node, an extra child node Y_k was added with a symmetrical CPT. To control $\lambda_{Y_k}(X_k)$ messages, $P(Y_k|pa(Y_k))$ was varied. Results were averaged over 100 instances. As results demonstrate (figures 6 and 7), the average error approaches 0 and convergence of IBP reaches 100% as $\lambda_{Y_j}(X) \to 0, 1$.

7 Related work and conclusions

Empirical study of the performance of belief propagation algorithm [6, 4] in different types of coding networks including Humming codes, low-density parity check, and turbocodes, demonstrated that accuracy of IBP is considerably better when noise level σ is low. Those results correlate very well with proposition 1 since lower noise level means that code nodes receive stronger support for one value from their observed children $\lim_{\sigma\to 0} \lambda(U_i) \to 0, 1$.

An investigation into the distribution of cycle lengths in coding networks has demonstrated that a node has a low probability (less than 0.01) of being in a cycle of length less than or equal to 10 [5]. Furthermore, the CPTs derived for edges with low Gaussian noise σ define very strong correlation between parent/child node values. Thus, observed child node will send quite strong support for the observed value to the parent. Both of the above observations combined with our empirical findings, provide an insight into excellent performance of IBP in coding networks. Coding networks have good parameters for two different factors influencing convergence and accuracy of IBP: large loop size and strong ϵ-support.

The work presented here has two novelties. First, it provides a direct analytical connection between loop size, root priors, and evidence support and the error in the posterior marginals computed by IBP. We derived an exact expression for the error value only for a special case of a node in a single-loop network without evidence. However, the empirical evidence leads us to the same conclusions as our analytical findings which indicates that the mechanics behind the performance of IBP in single-loop network and multiple-loop networks with or without evidence is the same.

The second novelty is extending well-known loop-cutset criteria that guarantees convergence and correctness of IBP in loopy networks when evidence nodes constitute a loop-cutset to instances where loop-cutset nodes are not observed, but receive an ϵ-support. The proposed ϵ-cutset criteria guarantees the convergence and certain degree of accuracy when ϵ is sufficiently small. The next step in our research is to devise means of estimating the threshold ϵ value that will guarantee the convergence of IBP and desired degree of accuracy in posterior marginals computed by IBP.

References

[1] A. Becker, R. Bar-Yehuda, and D. Geiger. Random algorithms for the loop cutset problem. In *Uncertainty in AI (UAI'99)*, 1999.

[2] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113:41–85, 1999.

[3] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, pages 353–366, 1989.

[4] B. J. Frey and D. J. C. MacKay. A revolution: Belief propagation in graphs with cycles. In *Neural Information Processing Systems*, 1997.

[5] X. Ge, D. Eppstein, and P. Smyth. The distribution of cycle lengths in graphical models for iterative decoding. Technical Report UCI-ICS 99-10, UCI, 1999.

[6] K. Kask I. Rish and R. Dechter. Empirical evaluation of approximation algorithms for probabilistic decoding. In *Uncertainty in AI (UAI'98)*, 1998.

[7] Y. Weiss K. P. Murphy and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in AI (UAI'99)*, 1999.

[8] F. R. Kschischang and B. J. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communications*, 16:219–230, 1998.

[9] S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[11] D.J.C. MacKay R.J. McEliece and J.-F. Cheng. Turbo decoding as an instance of pearl's belief propagation algorithm. *IEEE J. Selected Areas in Communication*, 1997.

[12] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation 12*, pages 1–41, 2000.

8 Appendix 1

As we mentioned previously, normalization is not required in computation of π and λ messages. Here, we will formally prove that posterior marginal beliefs computed by IBP are the same whether messages sent between nodes are normalized or not.

LEMMA **8.1** *Let X be a node in a Bayesian network to which we apply IBP algorithm. Let $\lambda_j^{n(t)}(X)$ and $\lambda_j^{(t)}(X)$ denote messages received by node X after iteration t from its child Y_j with and without normalization accordingly:*

$$\lambda_j^{n(t)}(X) = \alpha_j \lambda_j^{(t)}(X)$$

Let $\pi_X^{n(t)}(U_i)$ and $\pi_X^{(t)}(U_i)$ denote messages received by node X during iteration t from its parent U_i with and without normalization accordingly:

$$\pi_X^{n(t)}(U_i) = \beta_i \pi_X^{(t)}(U_i)$$

Let $\lambda_X^{(t+1)}(U_i)$ and $\pi_j^{(t+1)}(X)$ represent outgoing messages computed at node X from $\lambda_j^{(t)}(X)$ and $\pi_X^{(t)}(U_i)$ that were not normalized. Let $\lambda_X^{n(t+1)}(U_i)$ and $\pi_j^{n(t+1)}(X)$ represent outgoing messages computed at node X from normalized $\lambda_j^{n(t)}(X)$ and $\pi_X^{n(t)}(U_i)$. Then, following holds:

$$\lambda_X^{n(t+1)}(U_i) = \alpha \lambda_X^{(t+1)}(U_i) \tag{12}$$

$$\pi_X^{n(t+1)}(Y_j) = \beta \pi_X^{(t+1)}(Y_j) \tag{13}$$

where α and β are constants independent from the values U_i and X. Let $BEL^{n(t)}(X)$ represent a belief in X computed during iteration t from normalized messages. Let $BEL^{(t)}(X)$ represent a belief in X computed during iteration t from messages that were not normalized. Then, after normalizing:

$$BEL^{n(t)}(X) = BEL^{(t)}(X) \tag{14}$$

Proof.

Assume that $\lambda_X(X)$ is always normalized. Let us compute a λ message sent from X to its parent U_i:

$$\lambda_X^{n(t+1)}(u_i) = \sum_X \lambda_X(x) \prod_j \lambda_j^{n(t)}(x) \Big(\sum_{u_k, k\neq i} P(x|u) \prod_{k,k\neq i} \pi_X^{n(t)}(u_k) \Big) =$$

$$= \sum_X \lambda_X(x) \prod_j \alpha_j \lambda_j^{(t)}(x) (\sum_{u_k, k \neq i} P(x|u) \prod_{k, k \neq i} \beta_k \pi_X^{(t)}(u_k)) =$$

$$= \prod_j \alpha_j \prod_{k \neq i} \beta_k \sum_X \lambda_X(x) \prod_j \lambda_j^{(t)}(x) (\sum_{u_k, k \neq i} P(x|u) \prod_{k, k \neq i} \pi_X^{(t)}(u_k)) = \alpha \lambda_X^{(t)}(u_i)$$

where α is independent from the value u_i:

$$\alpha = (\prod_j \alpha_j)(\prod_{k \neq i} \beta_k)$$

Now, let us compute a π message sent from X to its child Y_j:

$$\pi_j^{n(t)}(x) = \lambda_X(x) \prod_{k \neq j} \lambda_k^{n(t)}(x) (\sum_u P(x|u) \prod_i \pi_X^{n(t)}(u_i))$$

$$= \lambda_X(x) \prod_{k \neq j} \alpha_k^{(t)} \lambda_k^{(t)}(x) (\sum_u P(x|u) \prod_i \beta_i \pi_X^{(t)}(u_i))$$

$$= \prod_{k \neq j} \alpha_k \prod_i \beta_i \lambda_X(x) \prod_{k \neq j} \lambda_k^{(t)}(x) (\sum_u P(x|u) \prod_i \pi_X^{(t)}(u_i)) = \beta \pi_j^{(t)}(x)$$

where β is independent from the value of X:

$$\beta = \prod_{k \neq j} \alpha_k \prod_i \beta_i$$

Define:

$$B^{(t)}(x) = \lambda_X(x) \prod_j \lambda_j^{(t)}(x) (\sum_u P(x|u) \prod_i \pi_X^{(t)}(u_i))$$

$$B^{n(t)}(X) = \lambda_X(x) \prod_j \lambda_j^{n(t)}(x) (\sum_u P(x|u) \prod_i \pi_X^{n(t)}(u_i))$$

$$= \prod_j \alpha_j \prod_i \beta_i \lambda_X(x) \prod_j \alpha_j \lambda_j^{(t)}(x) (\sum_u P(x|u) \prod_i \beta_i \pi_X^{(t)}(u_i)) = k B^{(t)}(x)$$

where

$$k = \prod_j \alpha_j \prod_i \beta_i$$

Belief in X computed during some iteration t:

$$BEL(t)(x) = \gamma \lambda_X(x) \prod_j \lambda_j(x) (\sum_u P(x|u) \prod_i \pi_X(u_i)) = \gamma B(X)$$

where

$$\gamma = (\sum_X B(X))^{(-1)}$$

$$BEL^{n(t)}(x) \gamma^n \lambda_X(x) \prod_j \lambda_j^n(x) (\sum_u P(x|u) \prod_i \pi_X^n(u_i) = \gamma^n B^n(X)$$

where

$$\gamma^n = (\sum_X B^n(X))^{(-1)} = k^{(-1)} (\sum_X B(X))^{(-1)} = k^{(-1)} \gamma$$

Then:

$$BEL^{n(t)}(x) = \gamma^n B^{n(t)}(X) = \gamma^n k B^{(t)}(x) = k^{(-1)} \gamma k (\gamma)^{(-1)} BEL^{(t)}(x) = BEL^{(t)}(x)$$

□

THEOREM **8.1 (Normalization)** *Iterative belief propagation algorithm applied to a Bayesian netowrk G computes the same posterior marginals at each iteration t wheather λ and π messages passed between the nodes are normalized or not.*

Proof.

In lemma 8.1 we proved that when incoming messages into node X differ only within constant factor, then outgoing message computed by node X will also differ only within a constant factor and computed beliefs will be the same after normalization.

Applying this consideration recursively starting at iteration 1, we can conclude that messages sent to node X during iteration t in the algorithm where λs and πs are never normalized are the same within constant factor as messages sent to node X during iteration t where all or some of the λs and πs are normalized (as long as activation order is the same). Therefore, posterior belief for node X computed during iteration t is the same weather messages are normalized or not.

□

9 Appendix 2

Proof.[lemma 4.1]

Let X be an observed node in a Bayesian network G. Let x_e be an evidence value of X. Then, $\lambda_X(x_e) = 1$ and $\lambda_X(x_i) = 0, x_i \neq x_e$. From equation 2, it follows that $\lambda^(t)(x_i) = 0, x_i \neq x_e$.

From equation 1, $\pi^{(t)}_{Y_j}(x_i) = 0, x_i \neq x_e$. After normalization, $\pi^{(t)}_{Y_j}(x_e) = 1$. Thus, the values of message $\pi^{(t)}_{Y_j}(X) = 1$ are constant and are independent from any messages that node X receives from its neighbors.

From equation 2:

$$\lambda^(_X t)(u_i) = \prod_j \lambda_j^{(t-1)}(x_e) \sum_{u_k \neq u_i} P(x_e|u) \prod_k \pi_k^{(t-1)}(u_k)$$

$$= k \sum_{u_k \neq u_i} P(x_e|u) \prod_k \pi_k^{(t-1)}(u_k)$$

where k is independent from value u_i or any value $\lambda_j^{(t-1)}(x_e)$. Therefore, $\lambda^(_X t)(u_i)$ does not depend on values of $\lambda_j^{(t-1)}(X)$.

□

Proof.[theorem 4.1]

Let X be an observed node in a Bayesian network G. Let G' be network obtained from G by creating a duplicate of node X^j for each child Y_j and then removing all edges between node X and its parents and replacing each edge (X, Y_j) with (X^j, Y_j) (X_j will also replace node X in the CPT of node Y_j). Posterior beliefs computed by IBP in G will be the same as posterior beliefs computed by IBP in G'due to lemma 4.1.

If observed nodes form a loop-cutset in G, then Bayesian network G' obtained from G by repeatedly applying procedure described above to each observed node X will be singly-connected. Since IBP always converges and computes correct posterior marginals in a singly-connected network, it will converge and compute correct posterior marginals for all nodes in G'. Since posterior beliefs computed by IBP in G will be the same as posterior beliefs computed by IBP in G' due to lemma 4.1, then IBP will converge and compute correct posterior marginals for all nodes in G.

□

Proof.[lemma 4.2]

Let us consider a leaf node X that is not observed. Let λ and π refer to the messages sent by IBP applied to G.

Since node X is not observed and does not have any children, $\lambda^{(t)}(X) = 1$ for all values of X. When IBP is applied to G, node X sends messages $\lambda_X(u_i)$ to each of its parents u_i:

$$\lambda_X^{(t+1)}(u_i) = \beta \sum_x \sum_{u_k, k \neq i} P(x|u) \prod_{k \neq i} \pi_x^{(t)}(u_k) \tag{15}$$

We can rewrite the equation above as:

$$\lambda_X^{(t+1)}(u_i) = \beta \sum_{u_k, k \neq i} (\sum_x P(x|u)) \prod_{k \neq i} \pi_x^{(t)}(u_k) \tag{16}$$

Since $\sum_x P(x|u) = 1$:

$$\lambda_X^{(t+1)}(u_i) = \beta \sum_{u_k, k \neq i} \prod_{k \neq i} \pi_x^{(t)}(u_k) \tag{17}$$

Since summation of the u_i for each π message yields a constant 1 we get that all values in the $\lambda_X(u_i)$ vector are 1.

If we remove node X from the network, belief propagation algorithm will compute the same messages sent between remaining nodes in the network and will generate the same posterior belief values. Node X is irrelevant to the computation of the posterior for all remaining nodes in the network.

Applying the same argument recursively yields the desired conclusion.

□

Proof.[theorem 4.2]

Repetitively find unobserved leaf nodes and remove them from the network. In the end, we will obtain a sub-network or a set of sub-networks G' whose leaves are observed. From Lemma 4.2, IBP applied to G is identical to IBP applied to G' as far as nodes in G' are concerned. If G' is singly connected or its observed variables constitute its loop cutset, IBP applied to G' is guaranteed to converge to the correct posterior marginals and therefore IBP applied to G computes the correct posterior for nodes in G'.

□

10 Appendix 3

LEMMA **10.1** *Let G be a single-loop Bayesian network as shown in figure 1. Assume all nodes are binary and there are no observed nodes in the loop. Let us define:*

$$S_0^k(B_k) = P(B_k|A=0) = \sum_{i=2}^{k-1} \prod_{i=2}^{k} P(B_i|B_{i_1}) P(B_1|A=0) \tag{18}$$

$$S_1^k(B_k) = P(B_k|A=1) = \sum_{i=2}^{k-1} \prod_{i=2}^{k} P(B_i|B_{i-1}) P(B_1|A=1) \tag{19}$$

Then:

$$diff_0^k = S_0^k(0) - S_1^k(0) = \prod_{i=1}^{k} (\beta_0^i - \beta_1^i) \tag{20}$$

$$diff_1^k = S_0^k(1) - S_1^k(1) = \prod_{i=1}^{k} (\beta_0^i - \beta_1^i) \tag{21}$$

Proof.

For $k=1$:

$$S_0^1(B_1) = P(B_1|A=0)$$

$$S_1^1(B_1) = P(B_1|A=1)$$

$$S_0^1(0) - S_1^1(0) = P(B_1=0|A=0) - P(B_1=0|A=1) = \beta_0^1 - \beta_1^1$$

$$S_0^1(1) - S_1^1(1) = P(B_1=1|A=0) - P(B_1=1|A=1) = (1-\beta_0^1) - (1-\beta_1^1) =$$

$$= -(\beta_0^1 - \beta_1^1)$$

Thus, $diff_0^1 = \beta_0^1 - \beta_1^1$ and $diff_1^1 = -(\beta_0^1 - \beta_1^1)$.

For $k=2$:

$$S_0^2(B_2) = \sum_{B_1} P(B_2|B_1) P(B_1|A=0)$$

$$S_1^2(B_2) = \sum_{B_1} P(B_2|B_1) P(B_1|A=1)$$

$$S_0^2(B_2) - S_1^2(B_2) = \sum_{B_1} P(B_2|B1)(P(B_1|A=0) - P(B_1|A=1)) =$$

$$= P(B_2|B_1=0)(\beta_0^1 - \beta_1^1) + P(B_2|B_1=1)((1-\beta_0^1) - (1-\beta_1^1)) =$$

$$= (\beta_0^1 - \beta_1^1)(P(B_2|B_1=0) - P(B_2|B_1=1))$$

$$S_0^2(0) - S_1^2(0) = (\beta_0^1 - \beta_1^1)(\beta_0^2 - \beta_1^2)$$

$$S_0^2(1) - S_1^2(1) = (\beta_0^1 - \beta_1^1)((1-\beta_0^2) - (1-\beta_1^2)) = -(\beta_0^1 - \beta_1^1)(\beta_0^2 - \beta_1^2)$$

Therefore, $diff_0^2 = (\beta_0^1 - \beta_1^1)(\beta_0^2 - \beta_1^2)$ and $diff_1^2 = -(\beta_0^1 - \beta_1^1)(\beta_0^2 - \beta_1^2)$.

Assume that claim is true for k:

$$diff_0^k = S_0^k(0) - S_1^k(0) = \prod_{i=1}^{k}(\beta_0^i - \beta_1^i)$$

$$diff_1^k = S_0^k(1) - S_1^k(1) = (-1)\prod_{i=1}^{k}(\beta_0^i - \beta_1^i)$$

Note that $diff_0^k = -diff_1^k$.

Let us compute $diff_0^{k+1}$ and $diff_1^{k+1}$:

$$S_0^{k+1}(B_{k+1}) - S_1^{k+1}(B_{k+1}) =$$

$$= \sum_{i=1}^{k}\prod_{i=2}^{k+1} P(B_i|B_{i-1})P(B_1|A=0) - \sum_{i=1}^{k}\prod_{i=2}^{k+1} P(B_i|B_{i-1})P(B_1|A=1) =$$

$$= \sum_{B_k} P(B_{k+1}|B_k)S_0^k(B_k) - \sum_{B_k} P(B_{k+1}|B_k)S_1^k(B_k) =$$

$$= \sum_{B_k} P(B_{k+1}|B_k)(S_0^k(B_k) - S_1^k(B_k)) =$$

$$= P(B_{k+1}|B_k=0)(S_0^k(0) - S_1^k(0)) + P(B_{k+1}|B_k=1)(S_0^k(1) - S_1^k(1) =$$

$$= P(B_{k+1}|B_k=0)(S_0^k(0) - S_1^k(0)) - P(B_{k+1}|B_k=1)(S_0^k(0) - S_1^k(0) =$$

$$= (P(B_{k+1}|B_k=0) - P(B_{k+1}|B_k=1))((S_0^k(0) - S_1^k(0))) =$$

$$= (P(B_{k+1}|B_k=0) - P(B_{k+1}|B_k=1))\prod_{i=1}^{k}(\beta_0^i - \beta_1^i)$$

$$diff_0^{k+1} = (\beta_0^{k+1} - \beta_1^{k+1})\prod_{i=1}^{k}(\beta_0^i - \beta_1^i) = \prod_{i=1}^{k+1}(\beta_0^i - \beta_1^i) \tag{22}$$

$$diff_1^{k+1} = ((1-\beta_0^{k+1}) - (1-\beta_1^{k+1}))\prod_{i=1}^{k}(\beta_0^i - \beta_1^i) = (-1)\prod_{i=1}^{k+1}(\beta_0^i - \beta_1^i) \tag{23}$$

By induction, proof is complete.

□

THEOREM **10.1** *Let G be a single-loop Bayesian network as shown in figure 1. Assume all nodes are binary and there are no observed nodes in the loop. Let us define:*

$$P(A) = (\epsilon, 1-\epsilon)$$

$$\beta_0^i = P(B_i = 0|B_{i-1} = 0), \beta_1^i = P(B_i = 0|B_{i-1} = 1)$$

$$\gamma_0^j = P(C_j = 0|C_{j-1} = 0), \gamma_1^j = P(C_j = 0|C_{j-1} = 1)$$

Then the error $\delta = G^*(D) - G(D)$ *in the posterior marginals of node D computed by Iterative Belief Propagation will be:*

$$\delta = (\epsilon - \epsilon^2)((-1)^{B_n}(-1)^{C_m} \sum_{B_n, C_m} P(D|B_n, C_m)) \prod_{i=1}^{n}(\beta_0^i - \beta_1^i) \prod_{j=1}^{m}(\gamma_0^j - \gamma_1^j) \quad (24)$$

Proof.

Consider a single loop Bayesian network (figure 1). Without evidence, all $\lambda(X) = (1, 1)$. Let us make following definitions:

$$S_0(B_n) = P(B_n|A = 0) = \sum_{B_i} \prod_{i=2}^{n} P(B_i|B_{i-1})P(B_1|A = 0) \quad (25)$$

$$S_1(B_n) = P(B_n|A = 1) = \sum_{B_i} \prod_{i=2}^{n} P(B_i|B_{i-1})P(B_1|A = 1) \quad (26)$$

$$S_0(C_m) = P(C_m|A = 0) = \sum_{C_j} \prod_{j=2}^{m} P(C_j|C_{j-1})P(C_1|A = 0) \quad (27)$$

$$S_1(C_m) = P(C_m|A = 1) = \sum_{C_j} \prod_{j=2}^{m} P(C_j|C_{j-1})P(C_1|A = 1) \quad (28)$$

Let $G(D)$ be exact posterior belief in D and $G*(D)$ be a posterior belief computed by IBP:

$$G(D) = \sum_{B_n, C_m} P(D|B_n, C_m) \sum_A P(B_n|A)P(C_m|A)P(A) =$$

$$= \sum_{B_n, C_m} P(D|B_n, C_m) \sum_A S_A(B_n)S_A(C_m)P(A) \quad (29)$$

$$G*(D) = \sum_{B_n, C_m} P(D|B_n, C_m)\pi_{B_n}\pi_{C_m} =$$

$$= \sum_{B_n, C_m} P(D|B_n, C_m)(\sum_A P(B_n|A)P(A))(\sum_A P(C_m|A)P(A)) =$$

$$= \sum_{B_n, C_m} P(D|B_n, C_m)(\sum_A S_A(B_n)P(A))(\sum_A S_A(C_m)P(A)) \quad (30)$$

Let us compute error in the posterior marginal of D:

$$\delta = G(D) - G*(D) = \sum_{B_n, C_m} P(D|B_n, C_m) \times$$

$$\times(\sum_A S_A(B_n)S_A(C_m)P(A) - (\sum_A S_A(B_n)P(A))(\sum_A S_A(C_m)P(A))) =$$

$$= \sum_{B_n, C_m} P(D|B_n, C_m)(\epsilon S_0(B_n)S_0(C_m) + (1-\epsilon)S_1(B_n)S_1(C_m) - \epsilon^2 S_0(B_n)S_0(C_m) +$$

$$+\epsilon(1-\epsilon)S_0(B_n)S_1(C_m)+\epsilon(1-\epsilon)S_1(B_n)S_0(C_m))-(1-\epsilon)^2S_1(B_n)S_1(C_m))=$$

$$=\sum_{B_n,C_m}P(D|B_n,C_m)(\epsilon S_0(B_n)S_0(C_m)+S_1(B_n)S_1(C_m)-\epsilon S_1(B_n)S_1(C_m)-$$

$$-\epsilon^2S_0(B_n)S_0(C_m)+\epsilon S_0(B_n)S_1(C_m)-\epsilon^2S_0(B_n)S_1(C_m)+\epsilon S_1(B_n)S_0(C_m)-$$

$$-\epsilon^2S_1(B_n)S_0(C_m)-S_1(B_n)S_1(C_m)+2\epsilon S_1(B_n)S_1(C_m)-\epsilon^2S_1(B_n)S_1(C_m))=$$

$$=\sum_{B_n,C_m}P(D|B_n,C_m)\times$$

$$\times(\epsilon S_0(B_n)S_0(C_m)-\epsilon^2S_0(B_n)S_0(C_m)-\epsilon S_0(B_n)S_1(C_m)+\epsilon^2S_0(B_n)S_1(C_m)-$$

$$-\epsilon S_1(B_n)S_0(C_m)+\epsilon^2S_1(B_n)S_0(C_m)+\epsilon S_1(B_n)S_1(C_m)-\epsilon^2S_1(B_n)S_1(C_m))=$$

$$=\sum_{B_n,C_m}P(D|B_n,C_m)((\epsilon-\epsilon^2)S_0(B_n)S_0(C_m)-(\epsilon-\epsilon^2)S_0(B_n)S_1(C_m)-$$

$$-(\epsilon-\epsilon^2)S_1(B_n)S_0(C_m)+(\epsilon-\epsilon^2)S_1(B_n)S_1(C_m))=$$

$$=(\epsilon-\epsilon^2)\sum_{B_n,C_m}P(D|B_n,C_m)\times$$

$$\times(S_0(B_n)S_0(C_m)-S_0(B_n)S_1(C_m)-S_1(B_n)S_0(C_m)+S_1(B_n)S_1(C_m))=$$

$$=(\epsilon-\epsilon^2)\sum_{B_n,C_m}P(D|B_n,C_m)(S_0(B_n)-S_1(B_n))(S_0(C_m)-S_1(C_m)) \qquad (31)$$

Using the result of lemma 10.1, we can rewrite the expression for the error:

$$\delta=(\epsilon-\epsilon^2)\sum_{B_n,C_m}P(D|B_n,C_m)(-1)^{B_n}\prod_{i=1}^{n}(\beta_0^i-\beta_1^i)(-1)^{C_m}\prod_{j=1}^{m}(\gamma_0^j-\gamma_1^j)$$

$$=(\epsilon-\epsilon^2)((-1)^{B_n}(-1)^{C_m}\sum_{B_n,C_m}P(D|B_n,C_m))\prod_{i=1}^{n}(\beta_0^i-\beta_1^i)\prod_{j=1}^{m}(\gamma_0^j-\gamma_1^j) \qquad (32)$$

□