# An Edge Deletion Semantics for Belief Propagation and its Practical Impact on Approximation Quality

**Arthur Choi** and **Adnan Darwiche**
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095
{aychoi,darwiche}@cs.ucla.edu

## Abstract

We show in this paper that the influential algorithm of iterative belief propagation can be understood in terms of exact inference on a polytree, which results from deleting enough edges from the original network. We show that deleting edges implies adding new parameters into a network, and that the iterations of belief propagation are searching for values of these new parameters which satisfy intuitive conditions that we characterize. The new semantics lead to the following question: Can one improve the quality of approximations computed by belief propagation by recovering some of the deleted edges, while keeping the network easy enough for exact inference? We show in this paper that the answer is yes, leading to another question: How do we choose which edges to recover? To answer, we propose a specific method based on mutual information which is motivated by the edge deletion semantics. Empirically, we provide experimental results showing that the quality of approximations can be improved without incurring much additional computational cost. We also show that recovering certain edges with low mutual information may not be worthwhile as they increase the computational complexity, without necessarily improving the quality of approximations.

## Introduction

The complexity of algorithms for exact inference on graphical models is generally exponential in the model treewidth (Jensen, Lauritzen, & Olesen 1990; Lauritzen & Spiegelhalter 1988; Zhang & Poole 1996; Dechter 1996; Darwiche 2001). Therefore, models with high treewidth (and no local structure, Chavira & Darwiche 2005) can be inaccessible to these methods, necessitating the use of approximate algorithms. Iterative Belief Propagation (IBP), also known as Loopy Belief Propagation (Pearl 1988; Murphy, Weiss, & Jordan 1999), is one such algorithm that has been critical for enabling certain classes of applications (e.g., Frey & MacKay 1997), which have been intractable for exact algorithms. We propose in this paper a new perspective on this influential algorithm, viewing it as an exact inference algorithm on a polytree approximation of the original network. The approximate polytree results from deleting enough edges from the original network, where the loss of each edge is offset by introducing free parameters into the approximate network. We show that the iterations of belief propagation can be understood as searching for values of these free parameters that satisfy intuitive conditions that we formally characterize.

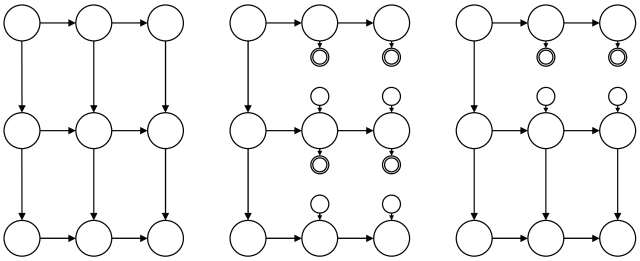


Figure 1: A network (left), a polytree approximation (center), and a more structured approximation (right).

This edge deletion semantics of IBP leads to a number of implications. On the theoretical side, it provides a new, intuitive characterization of the fixed points of IBP. On the practical side, it leads to a concrete framework for improving the quality of approximations returned by IBP. In particular, since IBP corresponds to deleting enough edges to yield a polytree, one wonders whether recovering some of the deleted edges can improve the quality of approximations; see Figure 1. The answer is yes, as we show later. Indeed, the edge recovery proposal can be quite practical if it leads to a multiply connected network that is still feasible for exact inference. This leads to another question: What edges are the most promising to recover? For this, we appeal to the semantics of edge deletion which suggest a criterion for recovering edges based on mutual information. We discuss the properties of this recovery method and provide empirical results, showing how it can identify a *small* set of edges that can effectively improve the quality of approximation without impacting much the complexity of inference. Our method also identifies edges whose recovery may increase inference complexity, without having a justifiable impact on the quality of approximations.

This paper is structured as follows. First, we start by formalizing approximate inference as exact inference on an ap-

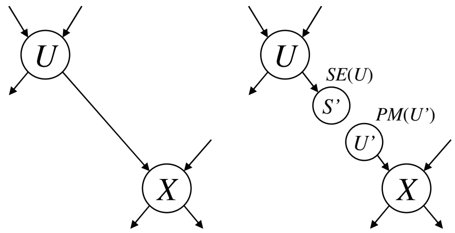


Figure 2: Deleting edge $U \rightarrow X$ by adding a clone $U'$ of $U$ and binary evidence variable $S'$.

proximate network which results from deleting edges. Second, we show how the deletion of edges introduces additional network parameters and propose a probabilistic semantics for choosing the values of these parameters. Third, we propose an iterative method for computing the characterized parameters, and show how belief propagation corresponds to the proposed framework when enough edges are deleted to yield a polytree. Fourth, we propose a criterion for recovering some of the original network edges, leading to an approximate network that is multiply connected. We finally present various empirical results to evaluate the edge recovery technique and discuss related work. Proofs for theorems appear in the Appendix.

## Deleting a Directed Edge

Let $U \rightarrow X$ be an edge in a Bayesian network, and suppose that we wish to delete this edge to make the network more amenable to exact inference algorithms. This deletion will introduce two problems. First, variable $X$ will lose its direct dependence on parent $U$. Second, variable $U$ may lose evidential information received through its child $X$. To address these problems, we propose to add two auxiliary variables for each deleted edge $U \rightarrow X$ as given in Figure 2. The first is a variable $U'$ which is made a parent of $X$, therefore acting as a clone of the lost parent $U$. The second is an *instantiated* variable $S'$ which is made a child of $U$, meant to provide evidence on $U$ in lieu of the lost evidence.[1] Note that auxiliary variable $U'$, as a clone, has the same values as variable $U$. Moreover, auxiliary variable $S'$ is binary as it represents evidence.

The deletion of an edge $U \rightarrow X$ will then lead to introducing new parameters into the network, as we must now provide conditional probability tables (CPTs) for the new variables $U'$ and $S'$. Variable $U'$, a root node in the network, needs parameters $\theta_{u'}$ representing the prior marginal on variable $U'$. We will use $PM(U')$ to denote these parameters, where $PM(u') = \theta_{u'}$. Variable $S'$, a leaf node in the network, needs parameters $\theta_{s'|u}$ representing the conditional probability of $s'$ given $U$. We will use $SE(U)$ to denote these parameters, where $SE(u) = \theta_{s'|u}$. Moreover, we will refer to $PM(U')$ and $SE(U)$ as *edge parameters.* Figure 3 depicts a simple network with a deleted edge, together with one possible assignment of the corresponding edge parameters.

We have a number of observations about our proposal for deleting edges:

- The extent to which this proposal is successful will depend on the specific values used for the parameters introduced by deleting edges. This is a topic which we address in the following section.
- If the deleted edge $U \rightarrow X$ splits the network into two disconnected networks, one can always find edge parameters which are guaranteed to lead to exact computation of variable marginals on both sides of the deleted edge; see Lemma 1 in the Appendix.
- The auxiliary variable $S'$ can be viewed as injecting a *soft evidence* on variable $U$, whose strength is defined by the parameters $SE(U)$. See the Appendix for some important observations on soft evidence and its properties. In particular, for queries that are conditioned on evidence $s'$, only the relative ratios of parameters $SE(U)$ matter, not their absolute values.

## Parametrizing Deleted Edges: ED-BP

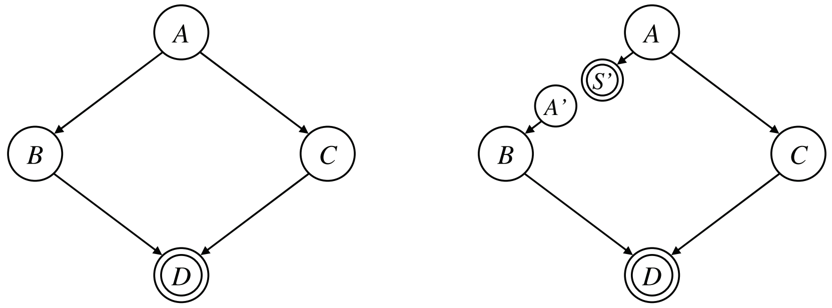
Given a network $N$ and evidence $\mathbf{e}$, our proposal is then to approximate this network with another $N'$ that results from deleting some number of edges $U \rightarrow X$ as given earlier. Moreover, when performing inference on network $N'$, we will condition on the augmented evidence $\mathbf{e}'$ composed of the original evidence $\mathbf{e}$ and each piece of auxiliary evidence $s'$ introduced when deleting edges. More formally, if $Pr$ and $Pr'$ are the distributions induced by networks $N$ and $N'$, respectively, we will use the conditional distribution $Pr'(.|\mathbf{e}')$ to approximate $Pr(.|\mathbf{e})$.

We cannot commence with the above proposal, however, without first specifying the parameters $PM(U')$ and $SE(U)$ for each deleted edge $U \rightarrow X$. There are two issues here. First, what parameters should we seek? Second, how do we compute them? As to the first question, a number of reasonable proposals can be extended, including the one for minimizing the KL–divergence between the original and approximate networks (Choi & Darwiche 2006). However, we propose to choose a less expensive criterion here, which has interesting properties. In particular, for each deleted edge $U \rightarrow X$, we require the following consistency conditions:

$$Pr'(u|\mathbf{e}') = Pr'(u'|\mathbf{e}'), \quad (1)$$
$$Pr'(u|\mathbf{e}' \setminus s') = Pr'(u'). \quad (2)$$

Condition 1 says that variables $U'$ and $U$ should have the same posterior marginals. This condition is mandated by the semantics of auxiliary variable $U'$, which say that $U'$ is a clone of variable $U$. Condition 2 is perhaps less obvious and best understood in the special case where deleting the edge $U \rightarrow X$ splits the network into two disconnected components, one containing $U$ (the $U$–component) and the other containing $X$ (the $X$–component). In this case, all evidence in the $X$–component will be disconnected from variable $U$. Condition 2 is then meant to ensure that this lost evidence is

[1] Our proposal for deleting an edge is an extension of the proposal given by (Choi, Chan, & Darwiche 2005), who proposed the addition of a clone variable $U'$ but missed the addition of evidence variable $S'$.



| $\Theta_{s'\|A}$ | $SE(A)$ |
|---|---|
| $\theta_{s'\|a}$ | 0.3437 |
| $\theta_{s'\|\bar{a}}$ | 0.6562 |

| $\Theta_{A'}$ | $PM(A')$ |
|---|---|
| $\theta_{a'}$ | 0.8262 |
| $\theta_{\bar{a}'}$ | 0.1737 |

Figure 3: A network $N$ (left), an approximate network $N'$ found after deleting $A \rightarrow B$ (center), along with parameters for auxiliary evidence variable $S'$ and clone $A'$ (right).

compensated for by the auxiliary evidence $s'$. In particular, Condition 2, in light of Condition 1, says that the impact of evidence $s'$ on variable $U$ is equivalent to the impact of all evidence on its clone $U'$:

$$\frac{Pr'(u|\mathbf{e}')}{Pr'(u|\mathbf{e}' \setminus s')} = \frac{Pr'(u'|\mathbf{e}')}{Pr'(u')}.$$

Assuming that an edge splits the network, only evidence in the $X$–component will have an impact on clone $U'$. Hence, in this case, the auxiliary evidence $s'$ does indeed summarize the effect of all evidence in the $X$–component.

Continuing with the assumption that an edge splits the network, it is interesting to note that the suggested conditions lead to exact results for marginals in the disconnected components. However, they will not necessarily lead to correct marginals over variable sets that cross components (see Lemma 1 in Appendix). Hence, these conditions lead to exact marginals over single variables when the deleted edge splits the network. If the edge does not split the network, not all evidence connected to $X$ will be disconnected from $U$. Hence, Condition 2 can be seen as overcompensating. It is indeed this observation which will be the basis of our heuristic for recovering edges, to be discussed later. Further, we will show that the stated conditions characterize the fixed points of IBP (and some of its generalizations) on Bayesian networks.

The question remains: How do we compute a parametrization which satisfies the above conditions? The following theorem provides an answer.

**Theorem 1** *Let $N$ be a Bayesian network and $N'$ be the result of deleting edges $U \rightarrow X$ from $N$. Conditions 1 and 2 hold in network $N'$ iff its edge parameters satisfy the following conditions:*

$$\begin{aligned} PM(u') &= Pr'(u|\mathbf{e}' \setminus s'), && (3) \\ SE(u) &= \alpha Pr'(\mathbf{e}'|u'), && (4) \end{aligned}$$

*where $\alpha$ is a constant $> 0$.*

Theorem 1 then suggests the following iterative method for identifying edge parameters. First, we start with a network $N'_0$ with a corresponding distribution $Pr'_0$, where all edge parameters $PM^0(U')$ and $SE^0(U)$ are initialized uniformly. For each iteration $t > 0$, the parameters for edges in $N'_t$ will be determined by performing exact inference on the approximate network $N'_{t-1}$ (from the previous iteration):

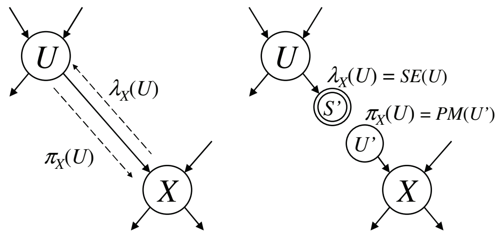


Figure 4: Correspondence between message passing in IBP and parameter updates in ED-BP.

$$\begin{aligned} PM^t(u') &= Pr'_{t-1}(u|\mathbf{e}' \setminus s'), && (5) \\ SE^t(u) &= \alpha Pr'_{t-1}(\mathbf{e}'|u'), && (6) \end{aligned}$$

where $\alpha > 0$ (we use $\alpha$ that normalizes $SE^t(U)$). We then keep iterating until all edge parameters converge to a fixed point (if ever).

We refer to the above method as ED-BP, an iterative method for parametrizing edge deletions based on belief propagation.[2] To justify the name, we will show that ED-BP subsumes IBP. In particular, for an edge $U \rightarrow X$ and an iteration $t$, we will use $\pi^t_X(U)$ to denote the message passed by IBP from parent $U$ to child $X$, and $\lambda^t_X(U)$ to denote the message passed from child $X$ to parent $U$.

**Theorem 2** *Let $N$ be a Bayesian network and $N'$ be the result of deleting $\underline{\text{every}}$ edge from $N$. Suppose we run IBP on $N$ and* ED-BP *on $N'$, where initially all $\pi^0_X(U) = PM^0(U')$ and all $\lambda^0_X(U) = SE^0(U)$. Then, for each edge $U \rightarrow X$ and each iteration t, we have:*

- *$\pi^t_X(U) = PM^t(U')$;*
- *$\lambda^t_X(U) = SE^t(U)$.*

*Moreover, for all variables $X$ in $N$ and each iteration t, we have $BEL_t(X|\mathbf{e}) = Pr'_t(X|\mathbf{e}')$.*

[2] See (Choi & Darwiche 2006) for a companion method, ED-KL, which is based on minimizing the KL–divergence.

This correspondence, depicted in Figure 4, shows that IBP is indeed searching for the edge parameters of an approximate, polytree network $N'$ obtained by deleting all edges from $N$. We note here that there is a spectrum of approximate networks that are polytrees, ranging from the polytree that has no edges (call it the minimal polytree), all the way to polytrees whose edges form a spanning tree of the network variables (call them maximal polytrees). The following theorem relates these different polytrees.

**Theorem 3** *Let N be a Bayesian network, $N'_m$ be the result of deleting every edge from N, and $N'$ be the result of deleting enough edges from N to yield a polytree. For every edge parametrization of $N'_m$ that satisfies Conditions 1 and 2, there is an edge parametrization of $N'$ that also satisfies these conditions and that leads networks $N'_m$ and $N'$ to agree on node marginals.*

That is, running ED-BP on the minimal polytree network versus a maximal one (or any approximation in between) yields the same approximations for node marginals. More precisely, it can be shown that the only difference between these approximate networks is that they lead to different message passing schedules in IBP. In particular, the minimal polytree approximation corresponds to a *parallel* message passing schedule, while maximal polytree approximations correspond to *sequential* schedules (Wainwright, Jaakkola, & Willsky 2003; Tappen & Freeman 2003).

It is important to note, however, that running ED-BP on more connected polytrees can yield more accurate approximations for marginals over multiple nodes. In general, computing such marginals is outside the scope of IBP, yet we will find joint marginals useful in the following section, when we recover deleted edges. We will therefore assume a maximal polytree approximation unless stated otherwise.

## Deciding Which Edges to Recover

Suppose we already have a polytree approximation of the original network, but we are afforded more computational resources. We can then relax the approximation by recovering some of the deleted edges. However, which edge's recovery would have the most positive impact on the quality of the approximation? Alternatively, we can ask: which edge's deletion had the most negative impact in the current approximation? To answer this question, let us consider again Conditions 1 and 2 which characterize the edge parameters that ED-BP searches for. Condition 1 requires that variable $U$ and $U'$ agree. Condition 2, given Condition 1, says that the soft evidence $s'$ on variable $U$ should summarize all evidence $\mathbf{e}'$ on its clone $U'$. This is meant to compensate for edge deletion which potentially disconnects the evidence now pertaining to $U'$ which formerly pertained to $U$. If all such evidence becomes disconnected from $U$, then Condition 2 is perfectly reasonable. However, if some of the evidence pertaining to $U'$ remains connected to $U$ in the approximate network, then the soft evidence on $U$ may be overcompensating. One way to measure the extent of this overcompensation is through the mutual information between $U$ and $U'$:

$$MI(U;U'|\mathbf{e}') = \sum_{uu'} Pr'(uu'|\mathbf{e}') \log \frac{Pr'(uu'|\mathbf{e}')}{Pr'(u|\mathbf{e}')Pr'(u'|\mathbf{e}')}.$$

**Algorithm 1** Static Edge Recovery (ER+S)

```
1: Given network $N$, find a maximal polytree $N_{BP}$ embedded in $N$; Run ED-BP on $N_{BP}$.
2: Rank deleted edges $U \rightarrow X$ based on $MI(U;U'|\mathbf{e}')$.
3: Recover the top $k$ edges into $N_{BP}$, yielding network $N'$; Run ED-BP on $N'$.
```

If deleting $U \rightarrow X$ splits the network into two disconnected subnetworks, then their mutual information is zero. Since edge deletion leads to exact results in this case, there is no point of recovering the corresponding edge. On the other hand, if $MI(U;U'|\mathbf{e}')$ is large, the interaction between $U$ and its clone $U'$ is strong in the approximate network and, hence, the soft evidence on $U$ may be overcompensating. Edge deletion may have therefore degraded the quality of our approximations, leading us to favor the recovery of such edges.

Algorithm 1 summarizes our proposal for edge recovery, which we refer to as ER+S. In Step 1, we choose an initial network $N_{BP}$ based on a maximal polytree embedded in the given network $N$, and run ED-BP to parametrize the deleted edges. In Step 2, we use the approximate network computed in Step 1 to compute the joint marginals $Pr'(uu'|\mathbf{e}')$ needed for the mutual information scores. We finally recover the top $k$ edges in Step 3 and run ED-BP again on a more connected network $N'$ to obtain more accurate approximations. In principle, we could repeat Steps 2 and 3 in an adaptive process, recovering in Step 3 a smaller number of edges and then re-ranking edges based on the improved approximation. We will refer to this method as ER+A and show in the following section that a static ranking, as used in ER+S, can often achieve similar performance even though it is less costly to compute.

Note that our heuristic requires the computation of joint marginals on pairs of variables, which are needed for obtaining the mutual information scores. Consider the fact that $Pr'(uu'|\mathbf{e}') = Pr'(u'|u,\mathbf{e}')Pr'(u|\mathbf{e}')$. We can score all edges outgoing from a particular node $U$ by applying exact inference to a polytree $N_{BP}$ once for each state $u$ of $U$: simply assert each state $u$ as evidence and retrieve the node marginals $Pr'(u'|u,\mathbf{e}')$. In this manner, Step 2 requires that we run the polytree algorithm as many times as there are instances $u$ of all tails $U$. If each variable has a constant number of values, and we have $n$ nodes in the network, then the number of runs needed is $O(n)$,

## Experimental Results

We start by observing the extent to which edge recovery improves the quality of approximations given by IBP, as measured by the average KL–divergence between exact and approximate marginals of unobserved variables.

Consider Figure 5, which depicts the improved approximations for the pigs and hailfinder networks.[3] The $x$–axes of these figures correspond to the number of edges recov-

[3] Most of the networks used for evaluation are available at http://www.cs.huji.ac.il/labs/compbio/Repository/.

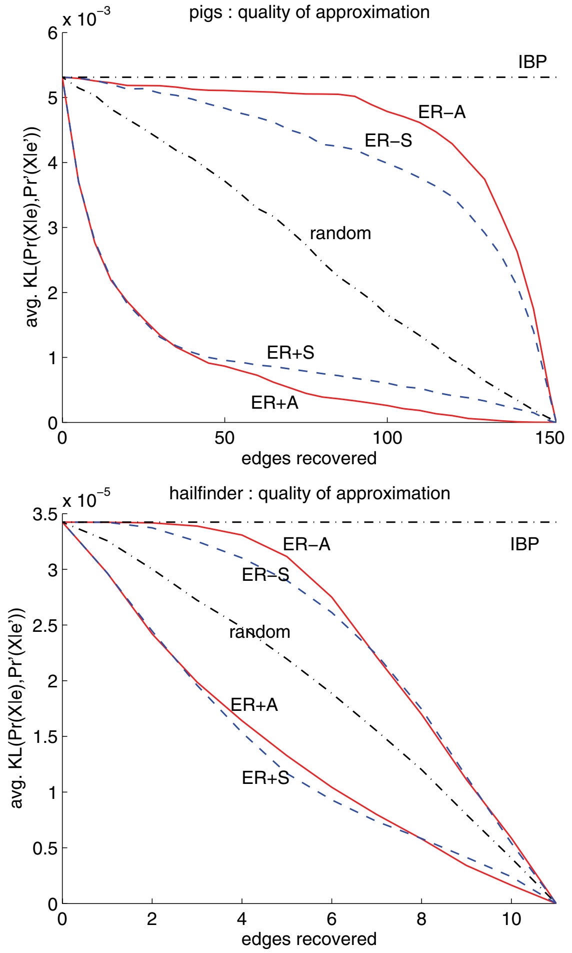


Figure 5: Quality of approximation vs edges recovered.

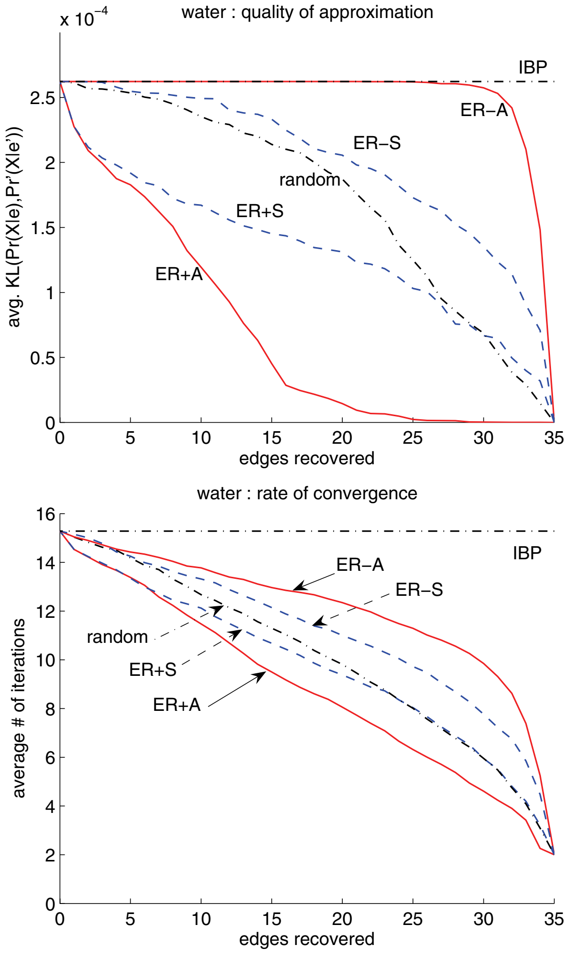


Figure 6: Quality and convergence rate vs edges recovered.

ered, starting from no edges recovered (a polytree approximation/IBP) to all edges recovered (original network/exact inference). Each point in these graphs is an average of 200 evidence instances, each fixing all leaf nodes according to the joint. All instances converged within 500 iterations, where convergence was determined when every parameter of an iteration is within $10^{-8}$ of the previous.

In these plots, we compared IBP and our edge recovery algorithm. We determined initial polytrees randomly, and used a jointree algorithm for exact inference in ED-BP. We then compared recovering edges with static rankings (ER+S), adaptive rankings (ER+A), and randomly. Further, we reversed the rankings (recovering edges with the lowest mutual information) for the non-random cases (ER-S, ER-A). In Figure 5, we see that in the pigs and hailfinder networks, static ranking performs closely to the more computationally intensive adaptive ranking. Moreover, in the pigs network, where we recovered edges in sets of five, we see substantial improvements in quality after only a few sets of edges are recovered. In the hailfinder network, where we recovered edges one at a time, the improvement is more modest. We can also see based on the reversed rankings that a substantial number of edges can be recovered without much benefit.

Consider Figure 6, which shows approximation quality and rates of convergence in the water network, where edges were recovered one at a time, and where we see a separation between static and adaptive ranking. For all ranking methods, we also see that the rate of convergence tends to improve linearly as more edges are recovered. It is important to note here that when recovering edges, we may increase network complexity, but we may also reduce convergence time. Hence, one must factor both issues when evaluating the costs incurred when recovering edges.

Figure 7 shows an interesting phenomena, where all ranking methods lead to better inference time over IBP when a small number of edges is recovered! This is due to two factors. First, the complexity of the jointree algorithm would often increase only gradually when recovering a small number of edges. Second, the improved convergence time may offset this increase in complexity, leading to better overall running time. A more detailed running time comparison is given in Table 1, which depicts the running times of our edge

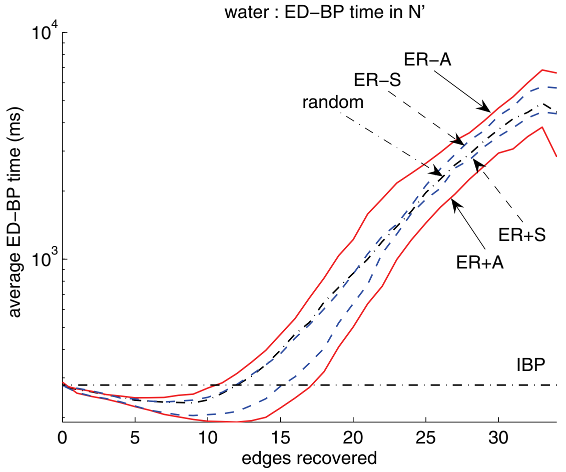


Figure 7: Inference time (only) versus number of edges recovered.

| network | Step 1 | Step 2 | Step 3 | Total |
|---|---|---|---|---|
| pigs | 1.778$s$ | 1.407$s$ | 1.452$s$ | 4.637$s$ |
| hailfinder | 0.078$s$ | 0.062$s$ | 0.074$s$ | 0.214$s$ |
| water | 0.278$s$ | 0.549$s$ | 0.235$s$ | 1.062$s$ |

Table 1: Average ER+S time spent at each step.

recovery method in its three stages, after recovering roughly 10% of the candidate edges.[4] Note that Step 1 corresponds to the running time of IBP, and Step 3 corresponds (roughly) to deleting edges randomly (no edge ranking is needed).

Table 2 summarizes the performance in these, and a few other networks, including a random $8 \times 8$ grid. Overall, we have observed that the quality of ED-BP approximations and rates of convergence tend to improve over IBP as more edges are recovered. Moreover, when we make a conscious choice in edges to recover, we can effectively improve the observed quality over random recovery, and further from IBP. Although our averages suggest a monotonic improvement in quality and convergence as more edges are recovered, we have indeed observed infrequent instances where recovering some edges has worsened both approximation quality and the convergence rate.

[4] Data collected on an Intel Xeon 1.50GHz CPU.

| | 0% | 10% recovered | | 50% recovered | |
|---|---|---|---|---|---|
| network | IBP | random | ER+S | random | ER+S |
| pigs | 5.31e-3 | 4.82e-3 | 2.21e-3 | 2.73e-3 | 7.88e-4 |
| hailfinder | 3.42e-5 | 3.26e-5 | 2.97e-5 | 2.20e-5 | 1.17e-5 |
| water | 2.62e-4 | 2.57e-4 | 2.03e-4 | 2.08e-4 | 1.40e-4 |
| win95pts | 7.77e-3 | 7.77e-3 | 3.05e-3 | 5.18e-3 | 1.73e-3 |
| $8 \times 8$ grid | 4.14e-5 | 4.00e-5 | 2.61e-5 | 2.85e-5 | 1.65e-5 |

Table 2: KL–divergence after recovering % of edges.

## Related Work

Iterative/loopy belief propagation has received a good deal of interest over the years as an approximate inference algorithm, particularly since revolutionary decoders for error correction have been shown to be instances of this algorithm (Frey & MacKay 1997). A number of formulations, generalizations, and alternatives to IBP have since been proposed (for references, see Yedidia, Freeman, & Weiss 2005). For example, similar characterizations of IBP also arise in expectation propagation (EP) (Minka 2001), where IBP corresponds to a disconnected network, and in tree-based reparametrizations (TRP) (Wainwright, Jaakkola, & Willsky 2003) where IBP corresponds to trees embedded in a complete *re*-parametrization of a distribution.

Among generalizations of IBP, those most relevant to edge deletion are perhaps generalized belief propagation (GBP) algorithms (Yedidia, Freeman, & Weiss 2005), to which EP and TRP also closely relate (Welling, Minka, & Teh 2005; Wainwright, Jaakkola, & Willsky 2003). Whereas IBP passes messages along edges in a network $N$, GBP algorithms pass messages according to an auxiliary structure composed of regions of $N$, where larger regions provide more accurate approximations. We sketch in the appendix how ED-BP can be simulated by a particular class of GBP algorithms based on joingraphs (Aji & McEliece 2001; Dechter, Kask, & Mateescu 2002). Therefore, our method identifies a new subclass of GBP, whose fixed points correspond to specific Bayesian networks $N'$ that approximate the original network $N$, and further whose semantics rely on more fundamental terms in graphical models.

Among generalizations of IBP, the design of approximating structures has received considerably less attention. The only proposal that the authors are aware of for the systematic design of approximations for IBP generalizations is the region pursuit algorithm for GBP (Welling 2004), which starts with an IBP approximation, identifies good candidates among a given set of proposed regions, and uses local message propagations to evaluate those that should be added to the approximation.

## Conclusion

We have proposed a method for approximate inference in Bayesian networks, which reduces the problem into one of exact inference on an approximate network obtained by deleting edges. We have shown how the influential IBP algorithm corresponds to an instance of our framework, where one deletes enough edges from the network to render it a polytree. The proposed framework and its subsumption of IBP leads to a number of theoretical and practical implications that we explored in this paper. On the theoretical side, our method leads to a new characterization of the fixed points of IBP. On the practical side, it leads to a practical method for improving the quality of IBP approximations by recovering those edges whose deletion is expected to worsen the quality of approximation the most.

## Acknowledgments

This work has been partially supported by Air Force grant #FA9550-05-1-0075-P00002 and JPL/NASA grant #1272258.

## Appendix: Proof Sketches and Other Results

### Proof of Theorem 1

$PM(u')$ parametrizes $U'$, so Equations 2 and 3 imply each other. Then we need to show that Equations 1 and 2 imply 4 and that Equations 3 and 4 imply 1. Note that given $U$, variable $S' \in \mathbf{E}'$ is independent of every other variable. Then from Equation 1 we have:

$$\begin{aligned} & Pr'(u'|\mathbf{e}') = Pr'(u|\mathbf{e}') \\ \iff \quad & Pr'(u'\mathbf{e}') = Pr'(u\mathbf{e}') = Pr'(\mathbf{e}'|u)Pr'(u) \\ & = Pr'(s'|u)Pr'(\mathbf{e}' \setminus s'|u)Pr'(u) \\ & = Pr'(s'|u)Pr'(u|\mathbf{e}' \setminus s')Pr'(\mathbf{e}' \setminus s') \\ & = \theta_{s'|u}Pr'(u')Pr'(\mathbf{e}' \setminus s') \text{ by Eq. 2 (3)} \\ \iff \quad & \theta_{s'|u} = Pr'(u'\mathbf{e}')/[Pr'(u')Pr'(\mathbf{e}' \setminus s')] \\ & = \alpha Pr'(\mathbf{e}'|u'), \end{aligned}$$

which is Equation 4, where $\alpha = [Pr'(\mathbf{e}' \setminus s')]^{-1}$. □

### Proof of Theorem 2

Suppose that the variable $X$ in network $N$ has parents $U_i$ and children $Y_j$. Then $X$ is the center of a star in $N'$ whose arms are auxiliary variables $U_i'$ and $S_j'$ introduced by deleting edges $U_i \to X$ and $X \to Y_j$, respectively. At iteration $t$, let $PM_i^t(U_i')$ parametrize $U_i'$ and $SE_j^t(X)$ parametrize $S_j'$. We first show by induction that for edges $X \to Y_j$, we have $\pi_{Y_j}^t(X) = PM_j^t(X')$, for all iterations $t$. Iteration $t = 0$ is given. Now assume for notation that all evidence $\mathbf{e}$ is virtual in $N$, and that the evidence in the star of $X$ in $N'$ is $\mathbf{e}_X'$. Then for any $t > 0$, we have:

$$\begin{aligned} & PM_j^t(x') = Pr_{t-1}'(x|\mathbf{e}' \setminus s_j') = Pr_{t-1}'(x|\mathbf{e}_X' \setminus s_j') \\ = \quad & \alpha Pr_{t-1}'(x\mathbf{e}_X' \setminus s_j') = \alpha \sum_{\mathbf{u}'} Pr_{t-1}'(x\mathbf{u}'\mathbf{e}_X' \setminus s_j') \\ = \quad & \alpha \sum_{\mathbf{u}'} \theta_{x|\mathbf{u}'} \prod_{u_i' \sim \mathbf{u}'} \theta_{u_i'}^{t-1} \prod_{s_k' \sim \mathbf{e}_X' \setminus s_j'} \theta_{s_k'|x}^{t-1} \\ = \quad & \alpha \sum_{\mathbf{u}'} \theta_{x|\mathbf{u}'} \prod_{u_i' \sim \mathbf{u}'} PM_i^{t-1}(u_i') \prod_{k \neq j} SE_k^{t-1}(x) \\ = \quad & \alpha \sum_{\mathbf{u}} \theta_{x|\mathbf{u}} \prod_{u_i \sim \mathbf{u}} \pi_X^{t-1}(u_i) \prod_{k \neq j} \lambda_{Y_j}^{t-1}(x) = \pi_{Y_j}^t(x), \end{aligned}$$

The other correspondences follow similarly. □

### Splitting a Network into Two Subnetworks

**Lemma 1** *Let $N$ be a Bayesian network and $N'$ be the result of deleting an edge $U \to X$ that splits $N$ into two disconnected subnetworks $N_U$ containing $U$ and $N_X$ containing $X$. Let $\mathbf{X}_U$ be the variables of subnetwork $N_U$ and $\mathbf{X}_X$ be the variables of subnetwork $N_X$. If the parameters $PM(u')$ and $SE(u)$ are determined by* ED-BP *in $N'$, then the marginal distributions for each approximate subnetwork are exact:*

$$Pr'(\mathbf{X}_U|\mathbf{e}') = Pr(\mathbf{X}_U|\mathbf{e}), \quad Pr'(\mathbf{X}_X|\mathbf{e}') = Pr(\mathbf{X}_X|\mathbf{e}).$$

**Proof** Let $u\mathbf{v}\mathbf{e}_U$ be an instantiation of variables and evidence in the set $\mathbf{X}_U$, and let $\mathbf{x}\mathbf{e}_X$ be an instantiation of variables and evidence in the set $\mathbf{X}_X$ (and thus $\mathbf{x}u\mathbf{v}\mathbf{e}$ is a complete instantiation in $N$). Since $Pr(\mathbf{x}_U|\mathbf{e}) = Pr(u\mathbf{v}|\mathbf{e})$ and $Pr'(\mathbf{x}_U|\mathbf{e}') = Pr'(u\mathbf{v}|\mathbf{e}')$, we have:

$$\begin{aligned} & Pr(u\mathbf{v}|\mathbf{e}) = \sum_{\mathbf{x}} Pr(\mathbf{x}u\mathbf{v}|\mathbf{e}) \propto \sum_{\mathbf{x}} Pr(\mathbf{x}u\mathbf{v}\mathbf{e}) \\ = \quad & \sum_{\mathbf{x}} Pr(\mathbf{x}\mathbf{e}_X|u\mathbf{v}\mathbf{e}_U)Pr(u\mathbf{v}\mathbf{e}_U) \\ = \quad & \sum_{\mathbf{x}} Pr(\mathbf{x}\mathbf{e}_X|u)Pr(u\mathbf{v}\mathbf{e}_U) = Pr(\mathbf{e}_X|u)Pr(u\mathbf{v}\mathbf{e}_U) \\ = \quad & Pr'(\mathbf{e}_X|u')Pr'(u\mathbf{v}\mathbf{e}_U) \propto \theta_{s'|u}Pr'(u\mathbf{v}\mathbf{e}_U) \\ = \quad & Pr'(u\mathbf{v}s'\mathbf{e}_U) \propto Pr'(u\mathbf{v}|s'\mathbf{e}_U) = Pr'(u\mathbf{v}|\mathbf{e}'). \end{aligned}$$

Similarly, to show $Pr(\mathbf{X}_X|\mathbf{e}) = Pr'(\mathbf{X}_X|\mathbf{e})$. □

Note again that when deleting $U \to X$ splits $N$ into two subnetworks, we do not not necessarily have correct marginals over variables that cross components.

### Proof of Theorem 3

Consider a minimal polytree network $N_m'$ that has been parametrized by ED-BP. The idea is that we want to recover from $N_m'$ only those polytree edges in $N'$. That is, we parametrize deleted edges in $N'$ using edges parameters from the corresponding edges in $N_m'$. Since $N'$ is a polytree, IBP is an exact inference algorithm on $N'$. Therefore, by Theorem 2, IBP messages propagated along the remaining polytree edges in $N'$ are exactly the edge parameters for the corresponding edges in $N_m'$.

Conversely, we can delete every edge in $N'$ to recover the minimal polytree network $N_m'$. The idea now is that deleting any individual edge in $N'$ splits the network into two disconnected subnetworks, and in particular, two polytrees. Since Lemma 1 tells us that the marginal distributions are unchanged in both subnetworks, the node marginals are unchanged when we delete a single edge in $N'$. Therefore, we can delete edges one at a time from $N'$ until we have the minimal polytree $N_m'$, without changing any of the node marginals. Moreover, when we delete a single edge from $N'$, the ED-BP conditions for edges already deleted in $N'$ remain satisfied. □

Running ED-BP in an arbitrary polytree $N'$ can be more concisely compared to running IBP in $N$ by specifying a message passing schedule. In particular, we first pass messages along the polytree edges of $N'$ in sequence, and then pass messages along deleted edges in parallel. When every edge is deleted, ED-BP in $N'$ corresponds to a fully parallel message passing schedule (see also Wainwright, Jaakkola, & Willsky 2003).

### On the Correspondence Between ED-BP and GBP

ED-BP can be simulated in terms of a GBP algorithm (Yedidia, Freeman, & Weiss 2005), and hence, an ED-BP fixed point corresponds to a stationary point of a free energy approximation. In particular, we shall sketch how an instance of ED-BP corresponds to an instance of iterative joingraph propagation (IJGP) (Aji & McEliece 2001; Dechter, Kask, & Mateescu 2002).

Let $N$ be a Bayesian network and $N'$ be the result of deleting edges $U \rightarrow X$ from $N$. To parametrize $N'$ using ED-BP, say we use a jointree algorithm for exact inference in $N'$. We take a suitably constructed jointree for $N'$ and add an edge between clusters containing $U'$ and $S'$, for each edge $U \rightarrow X$. This gives us a joingraph for IJGP.

We can then simulate ED-BP by IJGP in our specially constructed joingraph. Each IJGP iteration $t$ consists of two message-passing phases. First, we propagate message in the jointree embedded in our joingraph; this corresponds to propagating messages in a jointree for $N'_{t-1}$. Second, we propagate messages across the remaining edges connecting auxiliary variables $U'$ and $S'$; this corresponds to computing parameters $PM^t(u')$ and $SE^t(u)$ from $N'_{t-1}$ (see Equations 5 and 6).

## On Soft Evidence

There are two types of evidence that one may encounter: hard evidence and soft evidence. Suppose the variable $U$ takes on two states, $u$ and $\bar{u}$. Hard evidence on $U$ fixes $U$ to one of its two states, say $u$. Soft evidence on a variable $U$ is not as conclusive: it may increase our belief in $u$, but not to the point of certainty.

In a Bayesian network, we can model soft evidence on $U$ by adding an auxiliary child $S$ and asserting $s$ as hard evidence. The strength of soft evidence on $U$, which we denote by $SE(U)$, is then specified by the CPT parameters $\theta_{s|u}$ for each state $u$. For example, $S$ could denote a noisy sensor, evidence $s$ could denote a sensor reading, and $SE(U)$ could denote the reliability of the reading.

For example, say $SE(u) = 0.6$ and $SE(\bar{u}) = 0.2$. Then observing the auxiliary evidence $s$ has the following effect on the odds of $u$:

$$\frac{Pr(u|s)}{Pr(\bar{u}|s)} = \frac{Pr(s|u)}{Pr(s|\bar{u})}\frac{Pr(s)}{Pr(s)}\frac{Pr(u)}{Pr(\bar{u})} = \frac{SE(u)}{SE(\bar{u})}\frac{Pr(u)}{Pr(\bar{u})}.$$

In this example, the odds of $u$ after observing soft evidence is triple the odds of $u$ before. Note that the change in the odds of $u$ depends only on the relative ratios of parameters $SE(U)$, not their absolute values. Moreover, the effect of soft evidence on the global conditional distribution depends only on the change in the odds of $u$ (for details, see Pearl 1988, Chan & Darwiche 2005).

## References

Aji, S. M., and McEliece, R. J. 2001. The generalized distributive law and free energy minimization. In *Proceedings of the 39th Allerton Conference on Communication, Control and Computing*, 672–681.

Chan, H., and Darwiche, A. 2005. On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence* 163:67–90.

Chavira, M., and Darwiche, A. 2005. Compiling Bayesian networks with local structure. In *IJCAI*, 1306–1312.

Choi, A., and Darwiche, A. 2006. A variational approach for approximating Bayesian networks by edge deletion. Technical Report D–149, Computer Science Department, UCLA.

Choi, A.; Chan, H.; and Darwiche, A. 2005. On Bayesian network approximation by edge deletion. In *UAI*, 128–135.

Darwiche, A. 2001. Recursive conditioning. *Artificial Intelligence* 126(1-2):5–41.

Dechter, R.; Kask, K.; and Mateescu, R. 2002. Iterative join-graph propagation. In *UAI*, 128–136.

Dechter, R. 1996. Bucket elimination: A unifying framework for probabilistic inference. In *UAI*, 211–219.

Frey, B. J., and MacKay, D. J. C. 1997. A revolution: Belief propagation in graphs with cycles. In *NIPS*, 479–485.

Jensen, F. V.; Lauritzen, S.; and Olesen, K. 1990. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly* 4:269–282.

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

Minka, T. P. 2001. *A family of algorithms for approximate Bayesian inference*. Ph.D. Dissertation, MIT.

Murphy, K. P.; Weiss, Y.; and Jordan, M. I. 1999. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 467–475.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, California.

Tappen, M. F., and Freeman, W. T. 2003. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, 900–907.

Wainwright, M. J.; Jaakkola, T.; and Willsky, A. S. 2003. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory* 49(5):1120–1146.

Welling, M.; Minka, T. P.; and Teh, Y. W. 2005. Structured region graphs: morphing EP into GBP. In *UAI*.

Welling, M. 2004. On the choice of regions for generalized belief propagation. In *UAI*, 585. Arlington, Virginia: AUAI Press.

Yedidia, J.; Freeman, W.; and Weiss, Y. 2005. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory* 51(7):2282–2312.

Zhang, N. L., and Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *JAIR* 5:301–328.