Focused Belief Propagation for Query-Specific Inference

Anton Chechetka
Carnegie Mellon University

Carlos Guestrin
Carnegie Mellon University

Abstract

With the increasing popularity of large-scale probabilistic graphical models, even "lightweight" approximate inference methods are becoming infeasible. Fortunately, often large parts of the model are of no immediate interest to the end user. Given the variable that the user actually cares about, we show how to quantify edge importance in graphical models and to significantly speed up inference by focusing computation on important parts of the model. Our algorithm empirically demonstrates convergence speedup by multiple times over state of the art.

1 INTRODUCTION

Probabilistic graphical models (PGMs) have shown much success in modeling complex systems, from protein folding to sensor networks (Yanover and Weiss, 2002, Deshpande et al., 2004). Increasingly, applications require the use of *large-scale* graphical models. For example, a model of a relatively small academic department can have millions of factors (Richardson and Domingos, 2006). With such models, even relatively fast approximate inference techniques take many hours to finish. We argue that there exists a significant inference opportunity that remains unexploited by existing algorithms: often, *very few variables of the PGM are actually of interest to the end user.* In this paper, we show how exploiting information on which variables actually matter to the end user can dramatically speed up the inference.

In many real-life PGMs the majority of unknown variables serve only for modeling convenience, but do not directly affect the end user decisions. For example, in an automated system for patient monitoring (Beinlich et al., 1988), the only variable of direct interest may be whether the patient needs immediate attention of the hospital staff. In a smart home setting (Pentney et al., 2006), the variable of interest may be whether a certain room is likely to be occupied in the near future: to save energy, the smart home would turn the air conditioning off in rooms that are not likely to be occupied soon. However, an implicit assumption of the standard belief propagation (Pearl, 1988) variants, such as residual belief propagation (RBP) (Elidan et al., 2006) or residual splash BP (Gonzalez et al., 2009) is that every variable marginal is equally important. Therefore, a lot of computation is often wasted on refining beliefs over parts of the model that have very little effect on the query.

In this paper, we introduce a novel principled notion of importance of PGM edges to the query marginal. Unlike existing edge sensitivity estimates (Kjaerulff, 1993, Choi and Darwiche, 2008), our edge importance values can be computed efficiently. Based on our notion of edge importance, we propose an extension of RBP that focuses computation on the areas of the model that are likely to affect the inference result over the query the most. Moreover, computing edge importance can be done on-demand, eliminating the need to precompute importance values for all the edges in advance and preserving the anytime nature of BP. We show empirically on real-life large-scale graphical models that our algorithm, query-specific belief propagation (QSBP), converges several times faster than RBP.

2 BACKGROUND

We briefly review a particular formalism of probabilistic graphical models, namely factor graphs (for details, see Koller and Friedman, 2009), and loopy belief propagation, an approximate inference algorithm.

Probabilistic graphical models represent *factorized* probability distributions, where the distribution $P(\mathcal{X})$ over a large set of random variables $\mathcal{X}$ is decomposed into a product of low-dimensional functions:

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{f_\alpha \in \mathcal{F}} f_\alpha(\mathbf{X}_\alpha), \qquad (1)$$

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

where[1] every $\mathbf{X}_\alpha \subseteq \mathcal{X}$ is a subset of $\mathcal{X}$ (typically, $|\mathbf{X}_\alpha| \ll |\mathcal{X}|$), $f_\alpha \geq 0$ are **factors** and Z is the normalization constant. A probabilistic graphical model is a combination of the factorized distribution (1) and graphical structure induced by the factors f_α. Several alternative PGM formulations exist, depending on the type of graphs being used. Here, we use **factor graphs:** given the factorized distribution (1), the corresponding factor graph is a bipartite graph $(\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$ with one node for every factor $f_\alpha \in \mathcal{F}$ and every random variable $X_i \in \mathcal{X}$, and an undirected edge $(\alpha - i)$ for every pair of f_α and X_i such that $X_i \in \mathbf{X}_\alpha$ (see Fig. 1(a) for an example).

The central problem of this paper is, given the factor graph G and query variables $\mathbf{Q}$ to find the marginal distribution $P(\mathbf{Q})$. Unfortunately, this problem of *probabilistic inference* is known to be #P-complete in the exact case and NP-hard in the approximate case (Roth, 1996). Thus, we will address the problem of improving the convergence speed of belief propagation (Pearl, 1988), an approximate inference algorithm.

Loopy belief propagation (LBP), first proposed by Pearl (1988), is an algorithm for approximate inference in factor graphs, which has been very successful in practice (McEliece et al., 1998, Yedidia et al., 2003). Let Γ_α be the set of neighbors of node α in a factor graph. LBP is an iterative algorithm that repeatedly updates the messages $m_{\alpha - i}$ from factors f_α to their respective variables X_i until convergence, as follows:

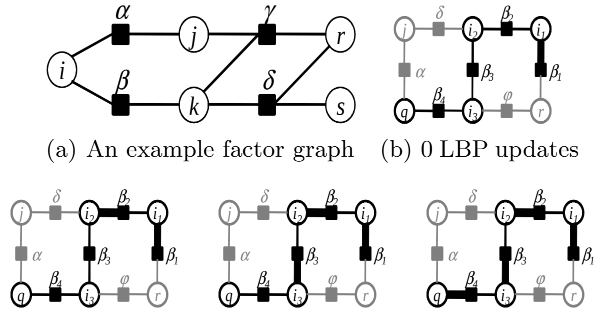
$$m^{(t+1)}_{\alpha - i}(x_i) \propto \sum_{\mathbf{x}_\alpha \setminus x_i} f_\alpha(\mathbf{x}_\alpha) \prod_{j \in \Gamma_\alpha \setminus i} \frac{\tilde{\mathbf{P}}^{(t)}(x_j)}{m^{(t)}_{\alpha - j}(x_j)}, \tag{2}$$

where $\tilde{\mathbf{P}}(\cdot)$ are the estimates of single-variable marginals defined as

$$\tilde{\mathbf{P}}(X_i = x_i) \propto \prod_{\alpha \in \Gamma_i} m_{\alpha - i}(x_i). \tag{3}$$

LBP is guaranteed to converge to exact variable marginals on graphs without cycles, but there are few guarantees for general factor graphs. Nevertheless, LBP beliefs are often successfully used instead of true marginals in applications (Yanover and Weiss, 2002).

Instead of directly computing an approximation to the query marginal $P(\mathbf{Q})$, loopy BP computes a set of single-variable marginals $\tilde{\mathbf{P}}(X_q)$. The query marginal is then approximated as $\tilde{\mathbf{P}}(\mathbf{Q}) \equiv \prod_{X_q \in \mathbf{Q}} \tilde{\mathbf{P}}(X_q)$.



(a) An example factor graph (b) 0 LBP updates

(c) 1 LBP update (d) 2 LBP updates (e) 3 LBP updates

Figure 1: An example factor graph (1(a)); A simple path $\pi = (\beta_1 - i_1 - \ldots - \beta_k - q)$ (solid black) in a factor graph. Thick edges indicate messages that become functions of $m_{\beta_1 - i_1}$ after the corresponding number of LBP updates.

Here, we are concerned with speeding up the convergence of LBP, so our goal is to recover $\tilde{\mathbf{P}}(\mathbf{Q})$ at a fixed point of LBP, instead of the true query marginal $P(\mathbf{Q})$.

Residual belief propagation (RBP). In the standard LBP, on every time step all the messages are recomputed and updated per the equation (2). However, many messages may change very little between the two time steps, and updating them often wastes computation. To improve efficiency, residual BP (Elidan et al., 2006) updates only one message per time step, namely the one that would have changed the most under LBP updates. More concretely, RBP maintains two messages for every edge, a current LBP message $m^{(t)}_{\alpha - j}$ and a new LBP message

$$\widehat{m}^{(t)}_{\alpha - i}(x_i) \propto \sum_{\mathbf{x}_\alpha \setminus x_i} f_\alpha(\mathbf{x}_\alpha) \prod_{j \in \Gamma_\alpha \setminus i} \frac{\tilde{\mathbf{P}}^{(t)}(x_j)}{m^{(t)}_{\alpha - j}(x_j)}. \tag{4}$$

The difference between the old and new messages is called the **residual** $r_{\alpha - i}$:

$$r_{\alpha - i} \equiv \| \widehat{m}^{(t)}_{\alpha - i} - m^{(t)}_{\alpha - i} \| \tag{5}$$

for some choice of norm. On time step $t+1$, RBP updates only one message, the one with the *largest residual.* That is, it sets $m^{(t+1)}_{\alpha - i} = \widehat{m}^{(t)}_{\alpha - i}$ for one edge, $\alpha - i = \arg\max_{\alpha - i} r_{\alpha - i}$ and keeps $m^{(t+1)}_{\beta - j} = m^{(t)}_{\beta - j}$ for all the other edges. After updating $m_{\alpha - i}$, the new messages $\widehat{m}_{\beta - j}$ only have to be recomputed if $\beta - i \in \mathcal{E}$. RBP thus avoids recomputing messages that change little, and significantly decreases the convergence time.

3 MEASURING IMPORTANCE OF MESSAGE TO THE QUERY

Consider again the factor graph in Fig. 1(a). Assume variable X_i is the query and on time step t message

[1] Notation: we will denote individual variables with regular font capital letters and index using Latin subscripts $(X_i, X_j, \ldots)$, sets of variables with bold font capital letters and index using Greek subscripts $(\mathbf{X}_\alpha, \mathbf{X}_\beta, \ldots)$ and denote the individual assignments with lower case letters $(X_i = x_i, \mathbf{X}_\alpha = \mathbf{x}_\alpha)$.

residuals are such that $r_{\delta-s} > r_{\gamma-j}$. What message should we update next? $m_{\delta-s}$ has a larger residual, but updating it will only change the belief over the irrelevant variable X_s and will not affect any other messages. Updating $m_{\gamma-j}$, on the other hand, would entail recomputing $\widehat{m}^{(t)}_{\alpha-i}$ and may change the query belief on the next step. Thus, one should choose updating $m_{\gamma-j}$ over $m_{\delta-s}$. However, the standard BP algorithms have no notion of the query $\mathbf{Q}$ and are unable to prioritize message updates by their importance to the convergence of $\tilde{\mathbf{P}}(\mathbf{Q})$: RBP will update the irrelevant $m_{\delta-s}$. To remedy this drawback of RBP, in this section we introduce a principled notion of message importance to the query and show how to compute these importance values efficiently. For simplicity, we will assume that there is only one query variable $\mathbf{Q} = \{X_q\}$, but our results generalize to multi-variable queries.

One can see from the LBP updates (2) that a change in message $m_{\alpha-i}$ propagates through the graph with consecutive LBP iterations: after one update, only the immediate neighbors of X_i are affected, then their respective neighbors and so on. For example, in Fig. 1(a), a change in message $m_{\delta-r}$ after one LBP update will affect $m_{\gamma-k}$ and $m_{\gamma-j}$, after the next update – $m_{\alpha-i}$, $m_{\beta-i}$, $m_{\delta-r}$ and $m_{\delta-s}$ and so on. Notice that the change in $m_{\delta-r}$ will impact the beliefs $\tilde{\mathbf{P}}(X_i)$ via different paths, namely $\delta-r-\gamma-k-\beta-i$ and $\delta-r-\gamma-j-\alpha-i$. Let us first quantify the importance of every such single path to the query belief $\tilde{\mathbf{P}}(X_q)$.

Consider a directed simple path $\pi = (\beta_1 \to i_1 \to \ldots \to \beta_k \to q)$ from factor β_1 to the query variable q in the full factor graph $G = (\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$. Fix the messages $\overrightarrow{m}_{-\pi}$ corresponding to all the edges not in π. Let us repeatedly apply LBP updates (2) to the messages corresponding to the edges in π. Then one can see that $m_{\beta_2-i_2}$ is a function of $m_{\beta_1-i_1}$, $m_{\beta_3-i_3}$ is a function of $m_{\beta_2-i_2}$ and so on. After $k-1$ LBP updates, m_{β_k-q} becomes a function of $m_{\beta_1-i_1}$ (see Fig. 1). Denote this function $m_{\beta_k-q} = F_\pi(m_{\beta_1-i_1}, \overrightarrow{m}_{-\pi})$. The sensitivity of m_{β_k-q} to changes in $m_{\beta_1-i_1}$ due to dependencies along path π is thus determined by the derivative

$$\left.\frac{\partial m_{\beta_k-q}}{\partial m_{\beta_1-i_1}}\right|_\pi \equiv \frac{\partial F_\pi}{\partial m_{\beta_1-i_1}} = \prod_{m=1}^{k-1} \frac{\partial m_{\beta_{m+1}-i_{m+1}}}{\partial m_{\beta_m-i_m}}. \qquad (6)$$

Maximizing (6) over all $\overrightarrow{m}_{-\pi}$, we bound from above the influence of m_{β_k-q} on $m_{\beta_1-i_1}$ via π. For tractability, we upper bound the factors of the product separately:

Definition 1. The **sensitivity strength** of a directed simple **path** $\pi = (\beta_1 \to i_1 \to \ldots \to q)$ is

$$\text{sensitivity}(\pi) = \prod_{m=1}^{k-1} \sup_{\overrightarrow{m}_{-\pi}} \left\| \frac{\partial m_{\beta_{m+1}-i_{m+1}}}{\partial m_{\beta_m-i_m}} \right\|. \qquad (7)$$

We adopt *log-dynamic range* as the message norm:

$$\|m_{\alpha-j}\| \equiv \log \frac{\max_{x_j} m_{\alpha-j}(x_j)}{\min_{x_j} m_{\alpha-j}(x_j)}, \qquad (8)$$

As Mooij and Kappen (2007) showed, for the norm (8) the suprema in (7) can be computed in closed form:

$$\sup_{\vec{m}} \left\| \frac{\partial m_{\alpha-i}}{\partial m_{\beta-j}} \right\| = \max_{x_i \neq x_i', x_j \neq x_j'} \tanh\left(\frac{1}{4} \log \frac{f_\alpha(x_i, \mathbf{x}_{\alpha\setminus\mathbf{i}}) f_\alpha(x_i', \mathbf{x}'_{\alpha\setminus\mathbf{i}})}{f_\alpha(x_i', \mathbf{x}_{\alpha\setminus\mathbf{i}}) f_\alpha(x_i, \mathbf{x}'_{\alpha\setminus\mathbf{i}})} \right). \qquad (9)$$

One can see that, if we keep the messages $\overrightarrow{m}_{-\pi}$ constant and change $m_{\beta_1-i_1}$ by Δ, the message m_{β_k-q} will eventually change by at most $\Delta \cdot \text{sensitivity}(\pi)$. In loopy models, however, there are typically many simple paths starting with $\beta_1 \to i_1$ and ending in q. Therefore, the effect of changing $\overrightarrow{m}_{-\pi}$ on the $\tilde{\mathbf{P}}(X_q)$ will be eventually transferred along many paths. It is possible in principle to bound the total effect of change propagation along *all the paths* in the graph G in polynomial time, for example adapting BP convergence analysis of Mooij and Kappen (2007). However, their approach relies on either inversion or eigenvalue computation over an $|\mathcal{E}| \times |\mathcal{E}|$ matrix, which typically is more expensive than running BP on the full model. Such a bound would thus be useless for fast query-specific inference. Instead, we choose the sensitivity of the *single strongest* directed *path* from an edge $(\alpha \to i)$ to the query q to be the importance value of that edge:

Definition 2. Given the query variable X_q, the **maximum sensitivity importance value** of an edge $(\alpha-i)$ is defined to be $A_{\alpha-i} \equiv \max_{\pi \in \Pi} \text{sensitivity}(\pi)$, where Π is the set of all directed simple paths that start with $\alpha \to i$ and have q as an endpoint.

3.1 COMPUTING EDGE IMPORTANCE

Even though we do not try to assess the cumulative influence of all the paths from $\alpha-i$ to q, Def. 2 still contains a maximization over all those paths. However, next we will show that the edge importance values of Def. 2 can still be computed efficiently. The main idea enabling the efficient computation is the same as in best-first search. Observe that first, the sensitivity strength of any path π decomposes over the edges of π (see Eq. 7), and second, from (9) it follows that $\sup_{\vec{m}} \left\| \frac{\partial m_{\alpha-i}}{\partial m_{\beta-j}} \right\| \in [0, 1)$, so for any path π' that is a subpath of π it holds that $\text{sensitivity}(\pi) < \text{sensitivity}(\pi')$. These two properties are analogous to those of a path length on a graph: path length decomposes into a sum of component edge lengths and increases as the path grows. Thus, similarly to a best-first search algorithm, which expands graph edges in the order of their

Algorithm 1: Edge importance computation

```
   Input: Factor graph $(\{\mathcal{X},\mathcal{F}\},\mathcal{E})$, query $q \in \mathcal{X}$.
1  $\mathcal{Q}$ is priority queue
2  foreach $(\alpha - i) \in \mathcal{E}$ do
3      if $i = q$ then $\mathcal{P}_{\alpha-i} \leftarrow 1$ else $\mathcal{P}_{\alpha-i} \leftarrow 0$
4      add $(\alpha - i)$ to $\mathcal{Q}$ w/priority $\mathcal{P}_{\alpha-i}$
5  while $\mathcal{Q} \neq \emptyset$ do
6      denote $(\beta - j)$ to be the top of $\mathcal{Q}$
7      foreach $(\gamma - k) \in \mathcal{Q}$ s.t. $(\beta - k) \in \mathcal{E}$ do
8          $\mathcal{P}_{\gamma-k} \leftarrow \max\left(\mathcal{P}_{\gamma-k}, \mathcal{P}_{\beta-j} \cdot \sup_{\vec{m}} \left\| \frac{\partial m_{\beta-j}}{\partial m_{\gamma-k}} \right\|\right)$
9      $A'_{\beta-j} \leftarrow \mathcal{P}_{\beta-j}$, remove $(\beta - j)$ from $\mathcal{Q}$
10 return $\vec{A}'$ - importance values for all the edges
```

shortest-path distance from the starting point, we can construct an edge importance computation algorithm (Alg. 1) that expands edges in the order of decreasing importance and as a by-product computes the exact importance values for every expanded edge.

Proposition 3. *Alg. 1 computes the exact maximum sensitivity importance values of Def. 2 for every edge $\alpha - i$, that is on line 9 of Alg. 1 $A'_{\beta-j} = A_{\beta-j}$.*

Proof sketch: By construction of edge priorities, at any time during execution of Alg. 1 for every edge $\alpha - i$ there exists a path $\pi = (\alpha \to i \to \cdots \to q)$ such that $\mathcal{P}_{\alpha-i} = \text{sensitivity}(\pi)$. Thus, the only possible failure mode of Alg. 1 is to get $A'_{\alpha-i} < A_{\alpha-i}$. The proof that such a failure is also impossible is by contradiction.

Denote $\mathcal{E}'$ the set of edges for which Alg. 1 returns $A'_{\alpha-i} < A_{\alpha-i}$. Let $\alpha - i$ be the edge from $\mathcal{E}'$ with the largest true importance $A_{\alpha-i}$. Consider the moment when $\alpha - i$ reaches the top of the priority queue $\mathcal{Q}$ and is expanded on line 6. The resulting importance weights $A'_{\beta-j}$ for all $\beta - j$ still in $\mathcal{Q}$ can be no greater than $A'_{\alpha-i}$ because $\sup_{\vec{m}} \left\| \frac{\partial m_{\beta-j}}{\partial m_{\gamma-k}} \right\| < 1$ and currently $\mathcal{P}_{\gamma-k} \leq \mathcal{P}_{\alpha-i} \forall \mathcal{P}_{\beta-j} \in \mathcal{Q}$. Therefore, all edges $\beta - j$ with $A_{\beta-j} \geq A_{\alpha-i}$ have already been expanded by Alg. 1. Denote $\pi^* = (\alpha \to i \to \beta \to j \cdots \to q)$ to be the largest sensitivity path from $\alpha - i$ to q. Because sensitivity$(\beta \to j \cdots \to q) >$ sensitivity(π^*), it follows that $A_{\beta-j} > A_{\alpha-i}$ and thus the edge $\beta - j$ has already been expanded by Alg. 1 with $\mathcal{P}_{\beta-j} = A_{\beta-j}$. Therefore, the correct value $\mathcal{P}_{\alpha-i} = A_{\alpha-i}$ should have been set on line 8 of Alg. 1 during the expansion of $\beta - j$, a contradiction.□

Proposition 4. *Suppose every factor of the factor graph $(\{\mathcal{X},\mathcal{F}\},\mathcal{E})$ contains at most d_f variables and every variable participates in at most d_v factors. Then the complexity of Alg. 1 is $O(d_f d_v |\mathcal{E}| \log |\mathcal{E}|)$.*

Proof sketch: Priority queue $\mathcal{Q}$ has size $|\mathcal{E}|$, so every update costs $O(\log |\mathcal{E}|)$. Every edge is expanded on lines 6-9 exactly once, and edge expansion entails updating priorities of at most $d_f d_v$ other edges.□

4 QUERY-SPECIFIC RESIDUAL BP

As Elidan et al. (2006) showed, BP convergence speed is significantly increased if the message updates are prioritized by message residual. They show that under some condition the distance between the current message $m_{\alpha-i}$ and the BP fixed point $m^*_{\alpha-i}$ is bounded from above by a monotonic function of the residual $r_{\alpha-i}$. Their algorithm, residual BP, can thus be viewed as greedily minimizing $\|\vec{m} - \vec{m}'\|_\infty$. RBP thus attempts to achieve uniformly small residuals over all the edges.

In a query-specific setting, however, every edge is only important to the degree that it influences the query belief $\tilde{\mathbf{P}}(X_q)$. As we have shown in the previous section, this importance can be quantified as maximum sensitivity edge weights $A_{\alpha-i}$. It is therefore natural to replace the L_∞ norm in RBP with weighted L_∞ norm:

$$\|\vec{m} - \vec{m}'\| \equiv \max_{\alpha-i \in \mathcal{E}} A_{\alpha-i} \|m_{\alpha-i} - m'_{\alpha-i}\|. \qquad (10)$$

The resulting algorithm, query-specific BP (QSBP, Alg. 2) prioritizes message updates in the order of their estimated impact on the query marginal. Observe that the changes from RBP, highlighted in the algorithm text, are minimal, and it is easy to modify an existing implementation of RBP to make it query-specific.

To illustrate how QSBP focuses the computation on the query, we show in Fig. 2 how message update counts are distributed for the two algorithms after the same running time on an image segmentation problem. The model in question takes an four-color image corrupted with Gaussian noise (top right corned of Fig. 2) and tries to recover the clean image in the top-left corner by penalizing disagreement between the neighboring pixels. One can see that QSBP concentrated message updates near the query and along the nearby color change boundary (the hardest part of the model), while RBP updates are distributed almost uniformly.

5 ANYTIME QSBP

One attractive property of the standard belief propagation algorithms is the fact that they are almost anytime: after message initialization ($O(|\mathcal{E}|)$ for LBP or $O(|\mathcal{E}| \log |\mathcal{E}|)$ for RBP), one can stop the algorithm at any point and read off the current estimate of the query belief according to (3). In contrast, QSBP has a relatively large startup cost: the initialization stage involves calling Alg. 1 with complexity $O(d_f d_v |\mathcal{E}| \log |\mathcal{E}|)$. In this section, we show how to

Algorithm 2: Query-specific belief propagation. Red underlined font denotes differences from RBP.

```
Input: Factor graph $G = (\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$, query $q \in \mathcal{X}$.
$\vec{A} \leftarrow$ Alg.1$(G, q)$ (find edge importance values)
foreach $(\alpha - i) \in \mathcal{E}$ initialize the message $m_{\alpha - i}$
foreach $(\alpha - i) \in \mathcal{E}$ compute $\widehat{m}_{\alpha - i}, r_{\alpha - i}$ using (4,5)
while not converged do
    $(\alpha - i) \leftarrow \arg\max_{\alpha - i}\left(r_{\alpha - i} \times A_{\alpha - i}\right)$
    $m_{\alpha - i} \leftarrow \widehat{m}_{\alpha - i}$
    foreach $\beta \in \mathcal{F}, j \in \mathcal{X}$ s.t. $(\beta - j), (\beta - i) \in \mathcal{E}$ do
        recompute $\widehat{m}_{\beta - j}$ and $r_{\beta - j}$ using Eq. 4, 5
return $\tilde{\mathbf{P}}(X_q)$ using Eq. 3
```

avoid expensive initialization, and thus make QSBP anytime, by interleaving edge weighting and inference.

5.1 SUBMODEL SELECTION

As a building block for our approach, we adopt a straightforward method of reducing inference complexity: instead of the full model $G = (\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$, use a submodel $G' = (\{\mathcal{X}', \mathcal{F}'\}, \mathcal{E}')$ such that $\mathcal{X}' \subseteq \mathcal{X}, \mathcal{F}' \subseteq \mathcal{F}, \mathcal{E}' \subseteq \mathcal{E}$. As a rule, the submodel G' is selected by breadth-first search (BFS) starting from the query variable and expanding up to a fixed radius (e.g., Pentney et al., 2006). However, BFS does not take into account the values of factors and the strength of influence of different edges.

Edge importance-based submodel selection. We propose an informed heuristic for submodel selection (Alg. 3) based on edge importance values of Def. 2: take the first k edges expanded by Alg. 1 and select the factors and variables of the submodel to be the endpoints of those expanded edges. Because Alg. 1 expands the edges in the order of decreasing importance, the resulting model is guaranteed to contain the k *most important* edges of G :

Proposition 5. *If* $(\{\mathcal{X}', \mathcal{F}'\}, \mathcal{E}')$*=Alg.3*$(G, q, k)$*, then* $\mathcal{E}'$ *contains the* k *edges from* $\mathcal{E}$ *with the highest* $A_{\alpha - i}$.

Proof sketch: As discussed in the proof sketch of Prop. 3, at the moment when edge $(\alpha - i)$ reaches the top of the priority queue $\mathcal{Q}$ in Alg. 1, we can guarantee for every $(\beta - i) \in \mathcal{Q}$ that $A_{\alpha - i} \geq A_{\beta - i}$. Thus Alg. 1 expands model edges in the order of decreasing $A_{\alpha - i}$. The proposition statement immediately follows.□

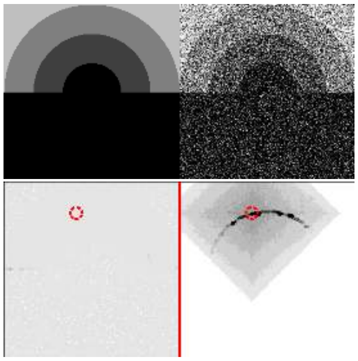


Figure 2: QSBP vs RBP on an image segmentation problem. From top-left corner, clockwise: clean image, noisy image, update counts for QSBP, update counts for RBP. Center of the red circle is the query. Darker shade means larger update frequency. White means no updates

Algorithm 3: Query-specific submodel selection

```
Input: Factor graph $G = (\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$, query $q \in \mathcal{X}$,
       size $k$.
$\mathcal{E}'' \leftarrow$ the first $k$ edges expanded by Alg. 1$(G, q)$
$\mathcal{X}', \mathcal{F}' \leftarrow$ variable and factor endpoints of $\mathcal{E}''$
$\mathcal{E}' = \{(\alpha - i) \mid \alpha \in \mathcal{F}', i \in \mathcal{X}', (\alpha - i) \in \mathcal{E}\}$
return $(\{\mathcal{X}', \mathcal{F}'\}, \mathcal{E}')$
```

Algorithm 4: Anytime QSBP (AQSBP)

```
Input: Factor graph $G = (\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$, query $q \in \mathcal{X}$
for $k = 1$ to $\lceil \log_2 |\mathcal{E}| \rceil$ do
    $G^{(k)} \leftarrow$ Alg. 3$(G, q, 2^k)$
    QSBP$(G^{(k)}, q)$ with (17) as convergence cond.
when stopped by the user, return current $\tilde{\mathbf{P}}(X_q)$
```

Edge importance invariance. Because edge importance in a model G depends on the existence of a high-sensitivity *path* π from that edge to the query, it is possible that in a submodel G' the same edge would have smaller importance if some component of π is absent from G'. Fortunately, Alg. 3 is guaranteed to preserve the importance of the k most important edges in the submodel G' that it selects:

Proposition 6. *Let* $G' = (\{\mathcal{X}', \mathcal{F}'\}, \mathcal{E}')$ *be the submodel selected by Alg. 3*$((\{\mathcal{X}, \mathcal{F}\}, \mathcal{E}), q, k)$*. Let also* $(\alpha - i)$ *be one of the* k *first edges to be expanded by Alg. 1. Denote* $A'_{\alpha - i}$ *to be the importance of* $(\alpha - i)$ *in* G' *according to Def. 2. Then* $A'_{\alpha - i} = A_{\alpha - i}$.

Proof sketch: Let $\pi = (\alpha \rightarrow i \rightarrow \beta_1 \rightarrow i_1 \rightarrow q)$ be the maximum sensitivity path for $(\alpha - i)$: sensitivity$(\pi) = A_{\alpha - i}$. Because $\sup_{\vec{m}} \left\| \frac{\partial m_{\beta - j}}{\partial m_{\gamma - k}} \right\| < 1$, for every subpath $\pi' = (\beta_m \rightarrow i_m \rightarrow \cdots \rightarrow q)$ we have that sensitivity$(\pi') >$ sensitivity(π). Thus $A_{\beta_m - i_m} > A_{\alpha - i}$ for every $(\beta_m - i_m) \in \pi$. From Prop. 5 it follows that every $(\beta_m - i_m) \in \pi$ will be also added to G' and $A'_{\alpha - i} \geq A_{\alpha - i}$. Observing that edge importance cannot increase in the submodel concludes the proof.□

5.2 ANYTIME INFERENCE

In practice, relatively small submodels selected by Alg. 3 or breadth-first search often provide a good approximation of the query distribution $\tilde{\mathbf{P}}(X_q)$ defined by the full model (Pentney et al., 2006). However, in Sec. 7 we demonstrate that for some real-life graphical models even large submodels give query marginals $\tilde{\mathbf{P}}(X_q)$ that are drastically different from those of the full model. It is thus undesirable in practice to restrict inference to a fixed submodel size: given enough time, all of the full model should be taken into account.

We want to combine the "best guess" behavior of small submodels early on in the inference process with computing the result corresponding to the full model if given enough time. A simple solution is to run inference on a sequence of submodels of increasing size (Alg. 4). A potential concern is that too much time will be wasted on the small submodels before moving on to larger ones. If we knew the available runtime at the outset, the optimal solution would be to select the largest possible submodel right away and spend all the time on inference in that submodel. However, because Alg. 4 grows submodels between iterations exponentially quickly, even without the benefit of knowing the available runtime in advance the loss in efficiency compared to the optimal action is small:

Proposition 7. *Let G_m to be the submodel selected by Alg. 3(G, q, m). Suppose the time $T(G')$ for QSBP to converge on G' is s.t. for every $G' \subseteq G'' \subseteq G$ it holds that $T(G') \leq T(G'')$. Then after time $\lceil \log_2 m \rceil T(G_m)$ Alg. 4 will start inference on a model $G' \supseteq G_m$.*

Proof sketch: Alg. 4 grows the submodels G' by a factor of 2 between iterations, so it needs $\log_2 m$ iterations to obtain $G' \supseteq G_m$. If the convergence times are nondecreasing with model size, every submodel smaller than G_m will take at most $T(G_m)$ to converge. □

One can see that Alg. 4 has constant initialization complexity and thus yields an anytime version of QSBP. Prop. 7 means that we can replace QSBP with its anytime version and incur at most a $\log(|\mathcal{E}|)$-factor penalty in runtime (under mild assumptions on the model G).

5.3 SUBMODEL CONVERGENCE CRITERION

Consider the error ε of the current belief $\tilde{\mathbf{P}}(X_q)$ of Alg. 4 with respect to the fixed point of belief propagation on the full model $(\{\mathcal{X}, \mathcal{F}\}, \mathcal{E})$, denoted $\widehat{\mathbf{P}}(X_q)$:

$$\varepsilon \equiv \|\tilde{\mathbf{P}}(X_q) - \widehat{\mathbf{P}}(X_q)\|. \qquad (11)$$

Denote also $\widehat{\mathbf{P}}^{(k)}(X_q)$ to be the fixed point of BP on the submodel $G^{(k)}$. The error ε can be split into two components, namely the difference between the two fixed points (called *approximation error* $\varepsilon_{\text{approx}}$):

$$\varepsilon_{\text{approx}} \equiv \|\widehat{\mathbf{P}}^{(k)}(X_q) - \widehat{\mathbf{P}}(X_q)\|, \qquad (12)$$

and the difference between the belief $\tilde{\mathbf{P}}(X_q)$ and the fixed point of the $G^{(k)}$ (called *inference error* $\varepsilon_{\text{infer}}$):

$$\varepsilon_{\text{infer}} \equiv \|\tilde{\mathbf{P}}(X_q) - \widehat{\mathbf{P}}^{(k)}(X_q)\|. \qquad (13)$$

By triangle inequality, we have

$$\varepsilon \leq \varepsilon_{\text{approx}} + \varepsilon_{\text{infer}}. \qquad (14)$$

Observe that $\varepsilon_{\text{approx}}$ is changed only when Alg. 4 switches to a larger model on line 2. On the other hand, $\varepsilon_{\text{infer}}$ only changes during inference on line 3. Therefore, to get good performance in practice, we need to trade off computation time spent by Alg. 4 between reducing the two sources of error. This tradeoff is controlled by convergence criteria for submodels $G^{(k)}$. Tight convergence criteria would entail spending more effort on reducing $\varepsilon_{\text{infer}}$ between attempts to change $\varepsilon_{\text{approx}}$, and vice versa for loose criteria.

Instead of using an arbitrary residual threshold as a convergence criterion for G', we use the following idea: suppose we had known the importance values $A_{\alpha-i}$ of all the edges $(\alpha - i) \in \mathcal{E}$ in advance. Then we want to run QSBP on a submodel G' *only so long as QSBP on the full model G would update the edges of G'*. The moment QSBP(G) would update an edge from $\mathcal{E} \setminus \mathcal{E}'$, we want to switch from G' to a larger submodel.

Importantly, we do not need to actually compute $A_{\alpha-i}$ for all the edges $(\alpha-i) \in \mathcal{E}$ to guarantee that the edges from $\mathcal{E} \setminus \mathcal{E}'$ would not be touched by QSBP yet at a certain point in time. To make such a guarantee, we rely on two observations. First, because Alg. 1 expands edges in the order of decreasing importance (Prop. 5), for any edge $\beta-j$ it holds that

$$A_{\beta-j} \leq \min_{(\alpha-i)\in\mathcal{E}''} A_{\alpha-i}, \qquad (15)$$

where $\mathcal{E}''$ is the set of edges obtained on line 1 of Alg. 3 during construction of submodel G'. Second, as Mooij and Kappen (2007) showed, for every edge $(\alpha-i) \in \mathcal{E}$ it holds that $r_{\alpha-i} \leq \|f_\alpha\|$. Therefore, for every edge $\beta-j \in \mathcal{E} \setminus \mathcal{E}'$ it holds that

$$r_{\beta-j} \cdot A_{\beta-j} \leq \max_{f_\gamma\in\mathcal{F}} \|f_\gamma\| \cdot \min_{(\gamma-k)\in\mathcal{E}''} A_{\gamma-k}. \qquad (16)$$

Observe that the maximum over all factors can be taken in $O(|\mathcal{F}|)$ time, which is no more then LBP initialization cost. As a result, we set the convergence condition for G' to be

$$\max_{(\alpha-i)\in\mathcal{E}'} r_{\alpha-i} A_{\alpha-i} \leq \max_{f_\alpha\in\mathcal{F}} \|f_\alpha\| \cdot \min_{(\alpha-i)\in\mathcal{E}''} A_{\alpha-i} \qquad (17)$$

One can show that the resulting anytime inference algorithm, which we call **AQSBP**, performs the same sequence of message updates as QSBP:

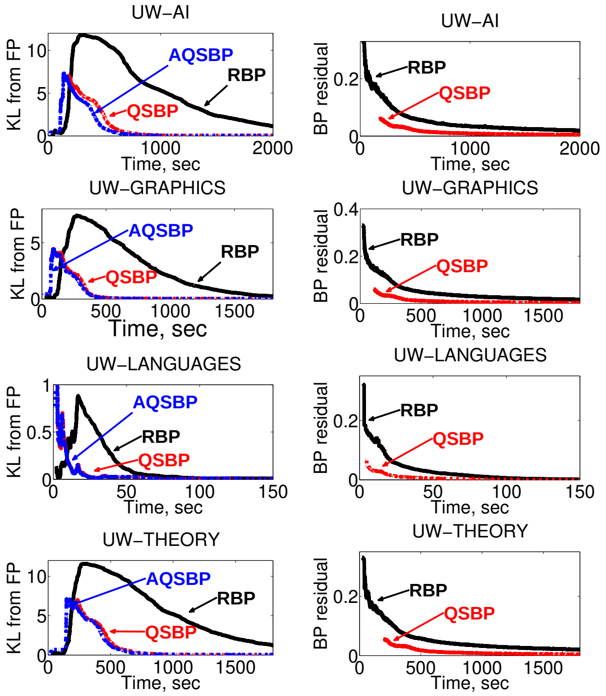


Figure 3: UW-CSE KL from fixed points (left column) and corresponding BP residuals (right column). Note the residuals are plotted on log scale.

Proposition 8. *Assuming the same message initialization, and using the messages at the end of QSBP run for $G^{(k)}$ to initialize the respective messages of $G^{(k+1)}$, the sequence of message updates performed by Alg. 4 is the same as for QSBP.*

The proof follows immediately from (16) and (17). It follows that interleaving edge weighting and inference has no effect on the end result even in the presence of multiple BP fixed points.

6 RELATED WORK

Query-specific inference in the existing literature is typically done by first selecting a query-specific submodel $G'(X_q)$ of the full factor graph G and then running a standard approximate inference algorithm, such as belief propagation. Query-specific submodel selection is usually done by breadth-first search, including all the variables and potentials within a certain radius from the query (Wellman et al., 1992, Pentney et al., 2006). Thus, neither submodel selection nor the inference stage of the existing approaches take into account the important information about relative importance of different edges to the query. In contrast, our approach incorporates such knowledge both at model selection stage and during inference. Pentney et al. (2007) used empirical mutual information between the query and every variable X_i as a measure of importance of X_i on the query belief during submodel selection, which has been shown to work well in practice, but requires access to the mutual information estimation for the underlying distribution. Our approach, in contrast, does not need any extra information besides the original graphical model G.

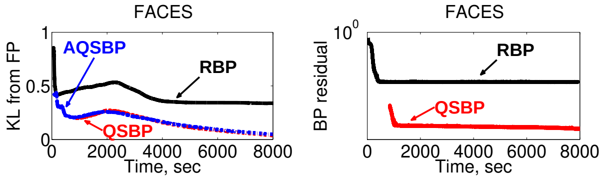


Figure 4: FACES KL from fixed points (left) and BP residuals (right) averaged over 10 folds and 30 single-variable queries.

Assessing the impact of a given edge $\alpha - i$ on the distribution $\mathbf{P}(X_q)$ defined by the graphical model G has been extensively studied in the field of *sensitivity analysis* in PGMs. For example, the effect of an edge removal on the query marginal can be computed with high accuracy (Kjaerulff, 1993, Choi et al., 2005, Choi and Darwiche, 2008, van Engelen, 1997) or exactly (Choi and Darwiche, 2008). However, those high accuracy approaches rely on inference in either reduced or full model, and are poorly suited for query-specific inference because of high computational cost. In contrast, our edge importance notion can be efficiently computed and interleaved with inference.

Finally, BP message analysis similar to ours can be used to bound the accuracy of LBP beliefs (Ihler, 2007, Mooij and Kappen, 2008), but that work does not address the question of speeding up inference. Incorporating these accuracy bounds into edge approximation values is an important direction of future work.

7 EXPERIMENTS

We have compared our algorithms, QSBP and AQSBP, with RBP on two relational graphical models: the UW-CSE Markov logic networks (Richardson and Domingos, 2006) and on a face recognition model (FACES) that “smoothes” the output of a single-image face classifier by introducing a factor favoring agreement between every two faces that have similar torsos. Torsos are defined as blobs of a fixed size directly under the face and are compared using standard color histogram distances. Our model improves recognition accuracy from 70% (standalone face recognition) to 87% on our dataset. We selected single-variable queries (30 for every dataset) such that the query beliefs take substantial time to converge. For every single-variable query, a separate BP run was performed and results were averaged. UW-CSE models vary in size from 10^3 variables and $5 \cdot 10^4$ factors (UW-LANGUAGES) to $7 \cdot 10^3$ variables and $5 \cdot 10^5$ factors (UW-GRAPHICS). FACES models have $2 \cdot 10^3$ variables and $9 \cdot 10^5$ factors. As the error measure we used the KL divergence from the BP fixed point on the full model $\widehat{\mathbf{P}}(X_q)$ to the current belief $\tilde{\mathbf{P}}(X_q)$. To judge convergence, we trace the values of maximum residual (importance-weighted for QSBP) as BP progresses. Small residuals indicate that the algorithm is close to convergence and vice versa.

Fig. 4 shows the results for FACES network averaged over 10 folds and 30 single-variable queries, with every fold representing an instantiation of the relational model on a different subset of the data. One can see that (A)QSBP consistently returned beliefs of significantly higher quality (several times smaller KL distance from the fixed point) than RBP throughout the inference process. Moreover, in the later stages of inference, (A)QSBP steadily decreased beliefs error as time progressed, while RBP beliefs error stagnated. QSBP has a large upfront cost of weighting all the model edges and takes 15 minutes to compute the first belief, but AQSBP finds a high-quality belief almost immediately. Outside of the difference in the initial stage, QSBP and AQSBP produce very similar results.

Fig. 3 shows the results for UW-CSE dataset. The evolution of the error over time for all three algorithms is qualitatively the same, but (A)QSBP converge more than twice as quickly as RBP. Due to the model structure, during inference, beliefs $\tilde{\mathbf{P}}(\mathbf{Q})$ significantly deviate from the fixed point before converging. Again, AQSBP produces first beliefs much faster than QSBP.

8 CONCLUSION

We addressed the problem of probabilistic inference in PGMs under the assumption that only few variables are of immediate interest for the user. Given the query set of important variables, we introduced a principled and efficiently computable notion of importance of PGM edges to the query. We showed how to use edge importance to focus inference on the parts of the model that are likely to affect query the most. By interleaving edge importance computations and inference, we preserved the anytime nature of belief propagation algorithm. Finally, we demonstrated empirically convergence speedups by several times compared to the state of the art RBP.

Acknowledgements: This work is supported in part by the ARO under MURI W911NF0710287 and W911NF0810242, by NSF Career IIS-0644225 and by Intel Research Fellowship. FACES model was developed jointly with Denver Dash and Matthai Philipose.

References

I. Beinlich, J. Suermondt, M. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probablistic inference techniques for belief networks. In *ECAIM*, 1988.

A. Choi and A. Darwiche. Focusing generalizations of belief propagation on targeted queries. In *AAAI*, 2008.

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

A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

G. Elidan, I. Mcgraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *UAI*, 2006.

J. Gonzalez, Y. Low, and C. Guestrin. Residual splash for optimally parallelizing belief propagation. In *AISTATS*, 2009.

A. Ihler. Accuracy bounds for belief propagation. In *UAI*, July 2007.

U. Kjaerulff. Approximation of Bayesian networks through edge removals. Technical report, Aalborg University, 1993.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

R. McEliece, D. MacKay, and J. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communications*, 16 (2):140–152, 1998.

J. Mooij and H. Kappen. Sufficient conditions for convergence of the sum-product algorithm. *IEEE Trans. on Inf. Theory*, 12(53), 2007.

J. M. Mooij and H. J. Kappen. Bounds on marginal probability distributions. In *NIPS*, 2008.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* 1988.

W. Pentney, A.-M. Popescu, S. Wang, H. A. Kautz, and M. Philipose. Sensor-based understanding of daily life via large-scale use of common sense. In *AAAI*, 2006.

W. Pentney, M. Philipose, J. A. Bilmes, and H. A. Kautz. Learning large scale common sense models of everyday life. In *AAAI*, 2007.

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2), 2006.

D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2), 1996.

R. A. van Engelen. Approximating Bayesian belief networks by arc removal. *IEEE Trans. on Patt. Analysis and Mach. Intelligence*, 19(8), 1997.

M. P. Wellman, J. S. Breese, and R. P. Goldman. From knowledge bases to decision models. *Knowledge Engineering Review*, 7:35–53, 1992.

C. Yanover and Y. Weiss. Approximate inference and protein folding. In *NIPS*, 2002.

J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium.* Morgan Kaufmann Publishers Inc., 2003.